Sensor Networks for Geospatial Event Detection — Theory and Applications

Thesis by

Annie Hsin-Wen Liu

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

California Institute of Technology

Pasadena, California

2013

(Defended May 21, 2013)

To Mom and Dad.

Acknowledgements

I am extremely fortunate to have had the privilege of working with and learning alongside my advisor Mani Chandy. Through his gentle guidance, I learned the joy of problem solving, the importance of communication, and that the ultimate purpose of a scientist is to apply science for the greater good. Mani has given me the freedom to explore, and I am forever grateful for the rich array of interesting research topics and people that I have come across in my studies.

I would like to express my special gratitude for several faculty: Andreas Krause, Rob Clayton, and Tom Heaton, whose insights and encouragements helped me identify interesting problems and make new discoveries; Richard Murray, who gave me the rewarding opportunity to teach an open-ended project course; and Adam Wierman and Jim Beck, for their insightful suggestions and discussions.

My doctoral journey has benefited from the guidance and hard work from many collaborators. Simon Labov and Karl Nelson at Lawrence Livermore National Laboratory helped me shape up ideas appropriate for the radiation application. Masumi Yamada and Jim Mori at Kyoto University shared with me their expertise on earthquake early warning. Julian Bunn lent me his expertise in numerical computing, in addition to his bright humor and high spirits. And finally, Richard, Monica, Ming-Hei, Leif, and Michael Aivazis of the CSN team, helped lay the foundation for this thesis.

This thesis would not have been possible without the help of my lab mates and friends. Michael always came to my rescue with the most superb technical support. I learned to balance work and play from Mihai and Matt Wu. Concetta and Jerome generously shared with me their experiences and wisdom. My good friends Mihoko, Ros, Wafa, Jean, and Gemma have kept me centered. Lastly, Matt Faulkner, who has become my closest friend and collaborator, never ceased to influence me with his intellect, curiosity, imagination, and thoughtfulness.

It would be hard to find a warmer or more welcoming environment than the CS department at Caltech. The never-ending efforts from all the faculty and administrators, especially Maria and Diane, have made my graduate student life as pleasant as possible. I would also like to thank Mathieu and Alain, in particular, for their genuine interest in my wellbeing and success.

Last but not least, I want to thank my family. My sister Jennifer has stood by me in every step, and I am especially grateful to my parents, Yeun-Wei Jen and Pang-Yuan Liu. Ten years ago, they made great efforts and sacrifices to create the opportunity for me to pursue my dreams, and they have continued to believe in me with unconditional love and support while I found my own path.

Abstract

This thesis presents theories, analyses, and algorithms for detecting and estimating parameters of geospatial events with today's large, noisy sensor networks. A geospatial event is initiated by a significant change in the state of points in a region in a 3-D space over an interval of time. After the event is initiated it may change the state of points over larger regions and longer periods of time.

Networked sensing is a typical approach for geospatial event detection. In contrast to traditional sensor networks comprised of a small number of high quality (and expensive) sensors, trends in personal computing devices and consumer electronics have made it possible to build large, dense networks at a low cost. The changes in sensor capability, network composition, and system constraints call for new models and algorithms suited to the opportunities and challenges of the new generation of sensor networks.

This thesis offers a single unifying model and a Bayesian framework for analyzing different types of geospatial events in such noisy sensor networks. It presents algorithms and theories for estimating the speed and accuracy of detecting geospatial events as a function of parameters from both the underlying geospatial system and the sensor network. Furthermore, the thesis addresses network scalability issues by presenting rigorous scalable algorithms for data aggregation for detection. These studies provide insights to the design of networked sensing systems for detecting geospatial events.

In addition to providing an overarching framework, this thesis presents theories and experimental results for two very different geospatial problems: detecting earthquakes and hazardous radiation. The general framework is applied to these specific problems, and predictions based on the theories are validated against measurements of systems in the laboratory and in the field.

Contents

List of Figures

Chapter 1

Introduction

This thesis presents theory, algorithms, simulations and experiments for detecting geospatial events using dense networks of noisy sensors. A geospatial event is initiated by a significant change in the state of points in a region in 3-D space over an interval of time. After the event is initiated, it may change the state of points over larger regions and longer periods of time. An example of such an event is an earthquake: it starts in a region of 3-D space (the hypocenter) in a relatively short interval in time (milliseconds), and then seismic waves propagate across large regions (hundreds of kilometers) over longer intervals of time (up to a few hundreds of seconds). Tsunamis, toxic plumes from nuclear plants, hurricanes, and wildfires can also be represented as geospatial events. Early detection of these events and accurate estimation of the event parameters, such as the magnitudes of earthquakes, gives the public time to respond appropriately.

Deploying and managing a networked sensing system for detecting geospatial events is a global challenge, for every continent has to deal with geospatial events. An analysis of the tradeoffs between network parameters helps in determining the appropriate designs for different problems and budgets, especially since many countries can benefit from rapid deployment of inexpensive sensors that are managed through community participation. My hypothesis is that there is a uniform, overarching framework for analyzing and designing a wide range of networks for responding to a large collection of geospatial events. In this thesis, I present a framework for geospatial event detection and parameter estimation, and I demonstrate that this common framework provides insights into designing networks for specific problems.

1.1 Contributions

I provide analyses for general models of geospatial events, and, in addition, I present detailed results for two specific problems: detecting earthquakes and hazardous radiation. Next, I briefly list my contributions in theory and experimentation, and then describe the overall problem in some detail.

Theory. I establish fundamental tradeoffs of detection and parameter estimation performance as a function of sensor and event parameters. I also present new algorithms for improving detection performance.

1. I present a general model of geospatial event detection. My contribution is the development of a unifying framework for analyzing a variety of problems that appear to be very different from each other. I use the framework for developing solutions to the problems of detecting earthquakes and radiation hazards. (Sec. 1.2.1 and Sec. 1.3.1).

2. Early warning of hazards gives people and machines more time to respond. Waiting to acquire more information about hazards that may be occurring allows more accuracy in determining whether a hazard is present, and in estimating the parameters of the hazard. The tradeoff between responding at a point in time or waiting to acquire more information is central to geospatial detection problems. I present a detailed analysis of this tradeoff. (Sec. 2.1 and Sec. 2.2).

3. Constraints, such as costs or location, may prevent the use of highly sensitive and accurate sensors. I present bounds on the accuracy of detection and localization (estimating the initial location) of geospatial events as a function of sensor and event parameters. These bounds provide insight about the types of sensors, the noise they are exposed to, and density of netowrk that are appropriate for an application. (Sec. 2.3 and Sec. 3.2.2).

4. Machine learning can reveal correlations among measurements of different sensors. Event detection and parameter estimation accuracy can be improved by exploiting these correlations. I present results that demonstrate the improvement obtained by using machine learning to estimate the correlations, and using the estimated correlations to detect events. (Sec. 6.4).

Applications and Experiments. I present the following experimental studies that validate the theoretical results and provide insight about intractable problems.

1. I used a laboratory testbed to measure the performance of a system that detects moving radiation sources. I compared experimental results with those from a model. (Sec. 4.1)

2. Bayesian inference provides a flexible framework for estimating event parameters; however, poor *a priori* estimates of parameters can only be overcome with extensive measurements. I suggest a heuristic for constructing prior estimates for a class of problems, and I validate the heuristic by comparing its results with measurements and simulations.

3. I used measurements of background radiation in a metropolitan region (downtown Sacramento, CA) to develop a measurement-based simulation to determine the performance of a system

with mobile sensors, such as police cars equipped with radiation sensors. I present results on the performance of such systems.(Sec. 4.2).

4. Estimating the parameters of an event is particularly difficult when spatially separated events occur almost concurrently. In such cases, many detection algorithms estimate parameters assuming a single event; parameters estimated in this way can be highly inaccurate. I present a scalable particle filter approach that estimates the parameters of almost concurrent earthquakes. The performance of this approach is evaluated with records following the Tohoku earthquake on March 11, 2011. (Chapter 5)

5. Machine learning to correlate measurements from sensors can help improve the signal to noise ratio of a sensor network. I present extensive experimental results of such a method to improve detection of a simulated virus outbreak in a real peer-to-peer network and detection of earthquakes in three seismic networks. (Sec. 6.4.4).

1.2 Geospatial Events

Definition. A geospatial event is initiated by a significant change in the state of points in a region in 3-D space over an interval of time. After the event is initiated it may change the state of points over larger regions and longer periods of time.

In addition to the earthquake example described earlier, consider the example of radiation detection. The problem of radiation detection may seem to fall completely outside of the framework of state-changes propagating across regions over time. Consider a radiation source introduced into a region at a point in time. Photons from the source travel at the speed of light in all directions, and hence the impact of the event appears to be almost instantaneous. The true impact of the event is, however, the absorbed dose over a period of time and not the instantaneous presence or absence of a photon. The absorbed dose (or a *rad* measure) increases over time.

This thesis focuses on events that are initiated by rapid changes in state within restricted regions and then propagate across larger regions. We study event propagation that can be accurately described as locus of state changes propagating outwards from a contiguous region in space-time. We do not consider events that are initiated by concurrent state changes over large regions. For example, climate change is initiated by changes at multiple points across the globe. As this does not fit the description of locus of state changes propagating from an interval in space-time, it is beyond the scope of this thesis.

In addition, we restrict attention to events that propagate over space in a continuous manner, as opposed to events that jump between non-contiguous regions. Events such as a cyber-attacks on networks that may be manifested by concurrent attacks at arbitrary locations are not considered.

Event detection. Whether a change in state of a region is significant and qualifies as an "event" depends on the application. For example, minor movements within the earth are not events for emergency first-responders, but still qualify as events for geophysicists. This thesis assumes that the definition of what is significant is a given input parameter.

This thesis does not consider methods for predicting the onset of events; we restrict attention to detecting the presence and behavior of events after they have been initiated. For example, the thesis does not deal with methods to predict future earthquakes; we do, however, study methods to estimate the future impact of earthquakes after they have started.

Event types. Most natural hazards can be represented as geospatial events. Hurricanes, earthquakes, and forest fires can be modeled, with some fidelity, as starting in some limited region and propagating across larger regions over time in a continuous manner. A model of geospatial events starting in a small region of space-time and expanding outwards is an approximation. For example, the genesis of an earthquake is stress built up over years over large tectonic plate boundaries; however, for the purposes of responding to an earthquake, it may be represented as propagating from a point in space-time.

In addition to natural hazards, some events caused by human action can also be modeled as geospatial events. For example, the emission of toxic plumes from the site of an accident, or water contamination from a factory can also be represented as geospatial events; however, events such as disease outbreaks do not fall naturally within our framework because airplane travel can transport germs across the globe, and diseases may appear to jump from one location to another.

1.2.1 Models of Geospatial Events

I next describe a simple model for geospatial events and then discuss its appropriateness and limitations for different problem domains.

Event source assumption. A geospatial event is initiated in some region of space-time. A common simplification is a point source model that assumes the event occurs at a single point, specified by the space-time coordinates (x_0, y_0, z_0, t_0). For example, an earthquake starts at a point in 3D space (x_0, y_0, z_0) (the hypocenter) at an instant in time t_0. Likewise, the model assumes that a forest fire starts with a spark at a specific location and time.

Associated with the event initiation are parameters (in addition to location and time) that describe the event. For example, the parameters that describe an earthquake include its magnitude; the parameters for a toxic plume include the concentration and type of toxic material; and, the parameters of a radiation hazard include the amount and type of radiation material and the shielding around the material. Let M be the set of parameters that describe the event. Then, an event initiation, for a point source, is described by the 5-tuple (M, x_0, y_0, z_0, t_0).

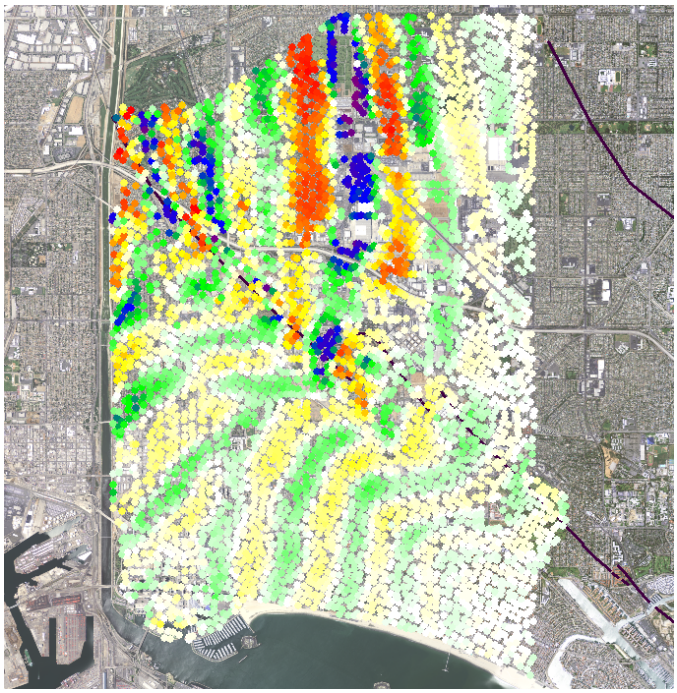


Figure 1.1: Visualizing $H(x, y, t)$ for a seismic event measured with a dense sensor network in Long Beach, CA. The event initiated in the upward left direction and propagates diagonally to the lower right corner. The different levels of impact experience at each geospatial point is color-coded for ease of visualization. [Source: Dr. Rob Clayton]

Event manifestation function. The manifestation of a geospatial event at any space-time point (x, y, z, t) is given by a vector $H(x, y, z, t)$ of environmental factors such as temperature, concentration of pollutants, and acceleration at point (x, y, z) in space at time t. The model of propagation of geospatial activity is specified by a function f that gives the manifestation H of the event at each point in space time, given an initiation (M, x_0, y_0, z_0, t_0) of a geospatial event:

$$H(x, y, z, t) = f(M, x_0, y_0, z_0, t_0, x, y, z, t) \tag{1.1}$$

The behavior of a geospatial event, initiated by a point source, is captured by the function f. Function f varies for different types of events. In the case of dispersion of a toxic plume, f describes conditions such a wind patterns, humidity, and precipitation. As an illustration, Fig. 1.1 shows a snapshot of vector H visualized during a seismic event measured by a dense network of seismic sensors deployed at Long Beach, CA[1].

The function f, for any realization of a propagating geospatial event, is drawn from a distribution F. For example, the speed of seismic waves depends on the geological structure of the material through which the waves propagate, and system designers may not know this structure precisely. We assume that if f is unknown, then designers know the distribution from which f is drawn.

[1]Data from Dr. Rob Clayton: `http://www.gps.caltech.edu/~clay/EQmovies/EQmovies.html`

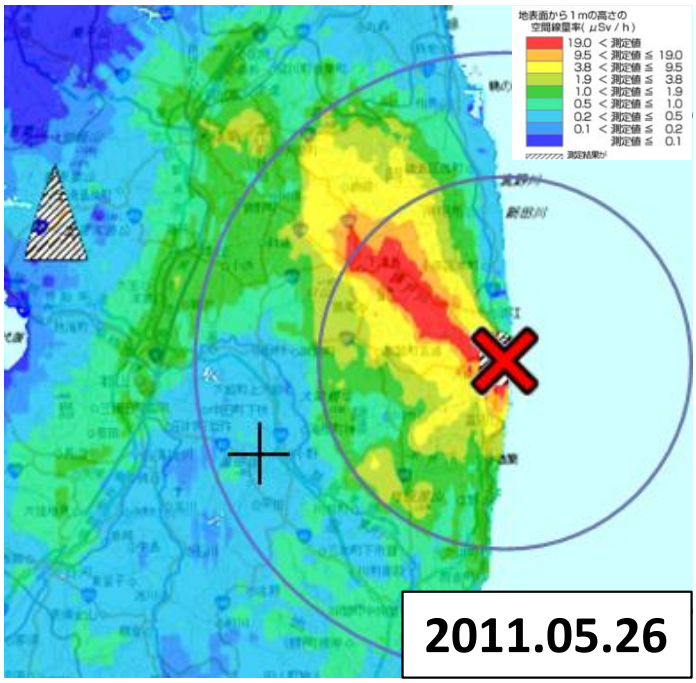


Figure 1.2: Distribution and intensity of radiation fallout reading following the Fukushima event on March 11, 2011. The map is created by air surveillance 1.5 month after the event [8].

1.2.2 Simple Event Models

Isotropic assumption. If the impact of the event is isotropic horizontally (invariant under horizontal rotation) and independent of the vertical dimension, then the propagation function f can be simplified to a function f' in which the locations of the source and impacted point are replaced by the horizontal distance between them:

$$H(x, y, t) = f'(M, d(x - x_0, y - y_0), t - t_0)$$

where $d(u, v)$ is the length of the 2-D vector (u, v).

Isotropic model is appropriate for some types of geospatial events. For example, signals (photons) from an unshielded radiation source in a clutter-free environment are emitted equally in all directions. In some applications, however, the event impact may propagate faster and with greater intensity in some directions than in others. An example is the emission of toxic plumes where the density and direction of the plume depend on wind speed. Another example is the radiation fallout following the Fukushima event. As shown Fig. 1.2, the distribution and intensity of the fallout readings are affected by wind and terrain (mountains). These complicating environmental factors are addressed by the manifestation function f Eq. (1.1).

Most results presented in this thesis are based on the isotropic model. Although this model may

not capture all the facets of a complex physical phenomenon, the results provide adequate accuracy for many problems, as validated by experiments. Analysis of this simple model also serve to provide insight into the interaction between parameters in geospatial event detection.

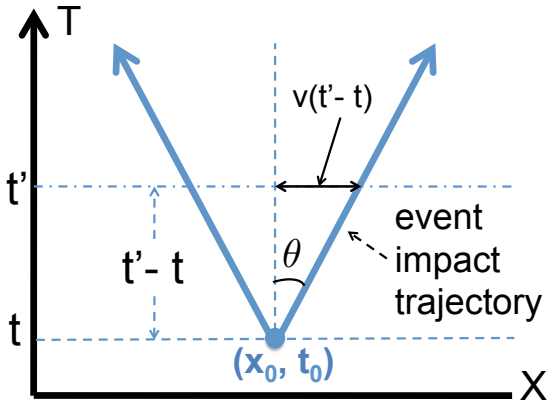
Space-time model. We first provide insight into the problem space by presenting a simple problem. Consider a 1-D isotropic event model with the 1-D space dimension on the x-axis and time t on the y-axis. Assuming that the event impact travels equally likely in all directions at speed v, an event initiated at a point (x, t) in space time will propagate along the space filled between two lines that resembles a cone with apex (x, t), where the boundary rays of the cone have an angle θ and $tan\theta = v$. The relationship is better visualized in the time-space plot in Fig. 1.3(a). In general, the speed v of propagation is not constant, and variable speeds give rise to non-conic shapes such as in Fig. 1.3(b).

In most cases, the impact of a geospatial event at a point in space continues for some time. For example, shaking from an earthquake may continue for many seconds and even minutes. Likewise, dangerous levels of radioactive material may remain at a point in space for days. Thus, the region of impact of a geospatial event is represented by the points between two cones if the velocities of arrival and departure of the impact are constant; more generally, the region of impact is represented by the space-time points between two upwardly increasing shapes as shown in Fig. 1.3(c).

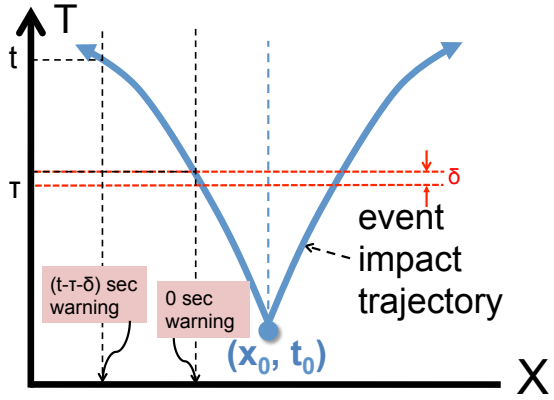
The magnitude of the impact of an event varies with time and distance from the initiation point. In many cases, such as for earthquakes, the magnitude of the impact usually decreases with distance from the initiation point. In other cases, for examples, forest fires, the intensity of the impact may change with time and distance in complex ways. Fig. 1.3(d) represents a situation where the intensity decreases with distance, with intensity shown by the color of the line.

There may be multiple events present. When multiple events occur close in space-time as shown in Fig. 1.3(e), the impact of the collection of events at a point is, in general, a complex function of the parameters of each of the events. This thesis at times uses a linear model that assumes that the impact of multiple events at a point is the sum of the impact of each of the events at that point, if each event was occurring independently. A linear model of impact is reasonable for detecting radiation hazards because the photon flux due to multiple sources is the sum of the fluxes from each source. Linear models may be inappropriate, however, for other applications such as earthquake detection in which seismic waves interfere in complex ways.

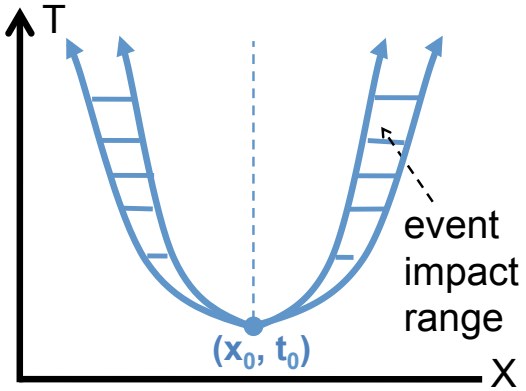
Event sources may move. For example, a vehicle containing hazardous radiation may travel on the roads. In such cases, the sensors detect a combination of signals from the same event. A moving event source can be modeled, with some fidelity, as a sequence of sources moving along some path. Fig. 1.3(f) shows the trajectory of a moving source and the wavefronts emanating from the source at different times. These wavefronts interact in complex ways.



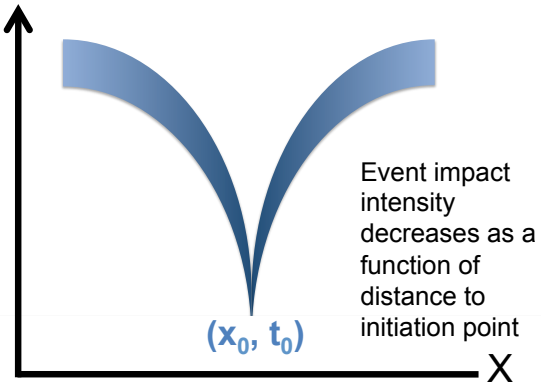
(a) Constant speed model



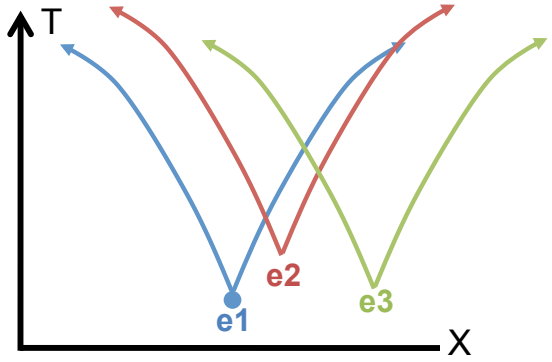
(b) Space-time dependent speed model. Speed may not be constant depending on environmental factors



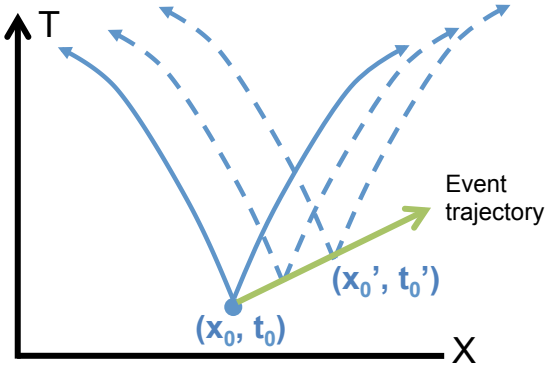
(c) Event Impact region. The event impact may last for some time. The impact region is the area between the event arrival and departure



(d) Impact attenuation. The event impact intensity usually decreases as function to the initiation point



(e) Multiple events. There may be multiple events occurring close in space-time. The sensors observe a complex combination of event signals



(f) Moving event sources. The sensors observe a different combination of signals from the same event

Figure 1.3: Geospatial event model illustrated in space-time plots. (x_0, t_0) is the event origin in space-time.

1.3 Models of Modern Sensor Networks

With advances in sensing technology and personal computing device, more and more new types of sensors are appearing in game consoles, cars, households, and smart phones. In comparison to traditional sensors, these modern sensors are generally inexpensive and small in size. Many sensor network projects are taking advantage of these alternative sensors for participatory community sensing. One of the earliest of such projects is the US Geological Survey *Did you feel it?* program that employs humans as sensors to construct surprisingly accurate shake-intensity maps from only subjective human input [7]. The use of *Twitter* tweets for detecting trending events [85] and geospatial events, such as earthquakes [79], have also been shown to be effective.

Challenges. There are three major challenges dealing with event detection using today's sensor networks.

1. **Volume**: A sensor network that employs smart phones or humans as sensors may include millions of sensors. As the network size grows, methods designed for small networks need to be replaced by scalable algorithms.

2. **Noise**: Inexpensive sensors are made possible with advances in sensing technology, e.g. \$50 CO_2 sensors for Arduino[2], \$40 standalone accelerometers[3], and free make-shift geiger counter that utilizes the camera on smart phones[4]. The level of noise, whether electronic or environmental, that these low-cost sensors experience can be orders or magnitude greater than that from high quality, and more expensive devices.

3. **Tradeoffs**: The design of geospatial sensor networks is a highly complex optimization problem. The problem becomes even more difficult with increasing varieties of sensors, communication mechanisms, and computing power.

1.3.1 Models of Sensors

We use a single model of sensors for different types of sensors and different types of applications. Consider a sensor of type s that experiences a manifestation H at time t measures a value $a_s(H)+\epsilon$, where ϵ is a random noise variable with zero mean. The function a_s maps true values to measured values. Ideally a_s is the identity function, and the sensor has zero noise. We discuss in later chapters how the sensor quality affects the sensor measurement function a_s. Here we first describe how sensor noise contributes to sensor errors $a_s(H) + \epsilon - H$.

Sensor noise can be *electronic* or *environmental*; and the resulting sensor errors can be categorized into three types:

[2] `http://sandboxelectronics.com/store/index.php?main_page=product_info&products_id=197`
[3] `http://www.phidgets.com/products.php?product_id=1053_0`
[4] `http://www.howtogeek.com/103184/your-android-phone-can-do-double-duty-as-a-geiger-counter`

1. **Detection error**: Electronic and environmental noise both contribute to false sensor events. For example, an accelerometer may detect an anomaly when a truck passes by. This is a valid detection of a local anomaly; however, this anomaly is not a geospatial anomaly of interest. This type of error is unavoidable with low quality sensors but can be made less frequent by setting sensor detection parameters.

2. **Location error**: Sensors may not be equipped with high quality GPS, or the location of the sensor may rely on self-reported information from sensor owners. These situations lead to error in sensor location that complicates the detection and estimation process.

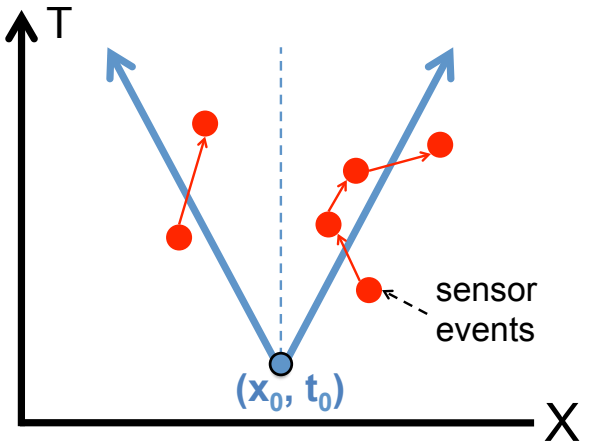
3. **Timing error**: The clocks in computing units and sensors in a network may not be perfectly synchronized. The clock drift cause sensors to misreport the time at which an anomaly occurs. This misreporting impacts system performance, as many detection and parameter estimation algorithms require that the times of sensor measurements to have microsecond precision.

To illustrate the impact of timing and location error, Fig. 1.4(a) shows the difference between the true impact of an event shown by the straight lines of a cone, and the measured impact shown by the dots identifying anomalies detected by sensors. Some sensors may not detect anomalies and the number of sensors near the source of the event may be small, and, as a consequence, the problem of identifying the geospatial event from the sensor readings (i.e. identifying the lines of the cone from the dots) can be difficult.

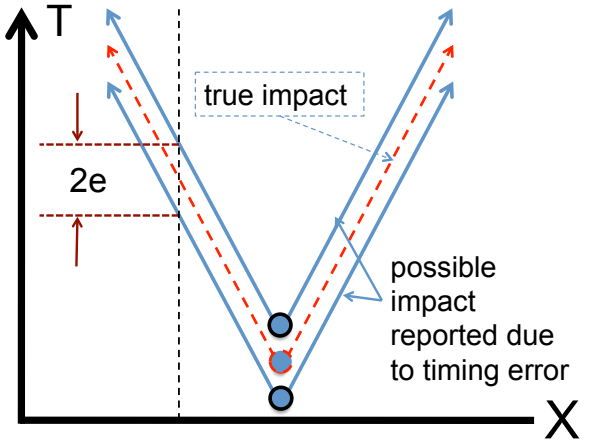
Let us represent errors in timestamps by a time interval $[-e, +e]$, where the probability of errors outside this interval are low. The timing error is represented by a band around the cone with vertical thickness $2e$, as shown in Fig. 1.4(b). In Fig. 1.4(c), sensor events that are true positives are shown as green while false positive are shown as red; the figure shows the difficulty of achieving high probabilities of true positives while ensuring low probabilities of false positives.

Communication model. Data from sensors can be streamed continuously to agents that fuse sensor data. Systems that stream sensor data must have adequate bandwidth to handle the traffic.

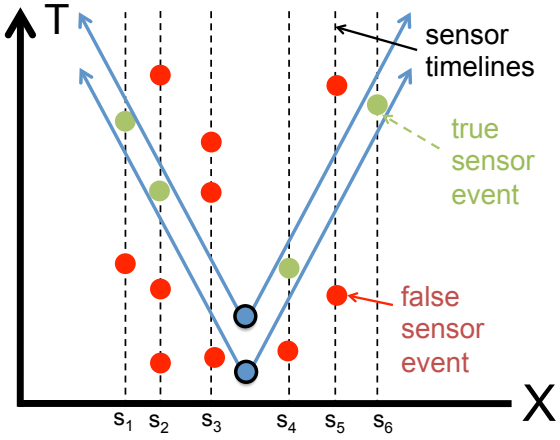
In the situation when the bandwidth and power supply do not permit a streaming model, raw sensor measurements can be compressed before they are sent. *Compressive sensing* is one of such methods that compresses the measurements by taking advantage of signal redundancy [13]. The original signal is then reconstructed at the server side. Such methods, however, require processing power at the sensors. In this thesis, we consider a simpler approach: the sensors send event messages to processing engines only when sensors detect events — local anomalies. An event message sent by the sensor has bits that identify the anomaly, the timestamp of the anomaly, and, possibly, measurements made during a short interval around the anomaly. The size and the frequency of event messages play a crucial role in the cost of the system, and on its efficacy.



(a) Identifying geospatial event is made difficult with errors in reported time and location.



(b) Sensor timing error leads to reduced accuracy in event parameter estimates.



(c) Timing error and sensor noise combined makes detection and parameter estimation more difficult

Figure 1.4: Sensor detection models illustrated in 1D space-time plots, assuming an isotropic event model. The event is modeled to initiate at a single point (x_0, t_0). The blue lines show the propagation of event impact in space and time. True sensor detections are green. False sensor detections are red.

1.4 Metrics for Performance Evaluation

Sensor networks for geospatial event detection have three main goals [47]:

1. Providing early warning so that people or machine components can react before a disaster hits; for example alerting people about impending intensive shaking from earthquakes.

2. Providing continuing situational awareness as a disaster unfolds; for instance, giving first responders information about which areas have been most badly damaged.

3. Providing data that is useful for scientific analysis, such as data about background radiation measured in an area over time.

Performance evaluation of such a system are summarized as metrics for: (1) warning, (2) ongoing situational awareness, or (3) science. The metrics for warning include the timeliness of the warning, and the accuracy of the estimation of the magnitude of the impact at each location and time. The metrics for ongoing situational awareness are similar, with the goal of detecting changes to an unfolding situation. Usually, first responders react in minutes, during which an elevator can be slowed down or a gas valve can be shut in seconds. Generally, constraints on the timeliness for situational awareness are less acute than for early warning. Science applications have even fewer constraints as it use data streams for data mining rather than event detection. For example, geologists are interested in tiny earthquakes that are not felt by anybody; the detection and characterization of such quakes can be carried out days after the event. In this thesis, we focus on applications dealing with warning and situation awareness. The most important criteria for early warning applications are *timeliness* and *accuracy*.

1.4.1 Timeliness

Warnings generated a short interval after the initiation of an event are based on data gathered over short times, and hence are more likely to be erroneous. The system can associate probabilities with warnings, generally giving higher probability warnings as time progresses. How fast the network can declare detection with high probability depends on the network and event parameters, such as the density of sensors, and the speed of the event propagation. It also depends on the data aggregation algorithm.

Timely detection leads to faster warning. As an example, Fig. 1.3(b) shows an event detection at a time τ. Alerts about the event are sent electronically to the impacted region, and as a first approximation, we assume that alerts reach all points in the region after a delay of δ seconds. The warning time for a point x in space is the delay between the instant $(\tau + \delta)$ and the instant at which dangerous intensities are experienced at point x. As shown in Fig. 1.3(b), some locations may have

adequate warning while other locations do not. One of the metrics for evaluating the effectiveness of system is the amount of area, or more appropriately, the fraction of the population in the region, that gets adequate warning time.

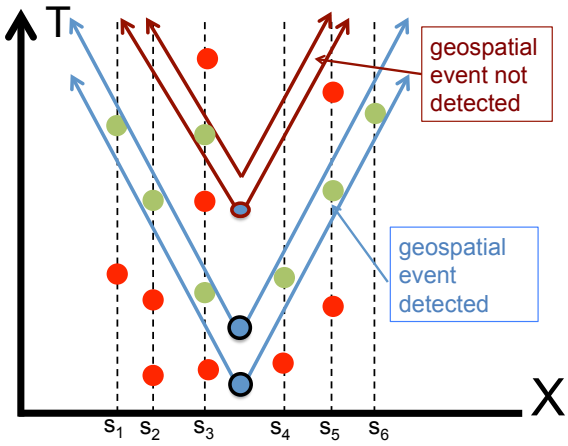
1.4.2 Accuracy

The accuracy of both *detection* and *parameter estimation* are also affected by event and network parameters. Next, we discuss issues that impact the accuracy and speed of geospatial event detection in terms of simple geometrical concepts.

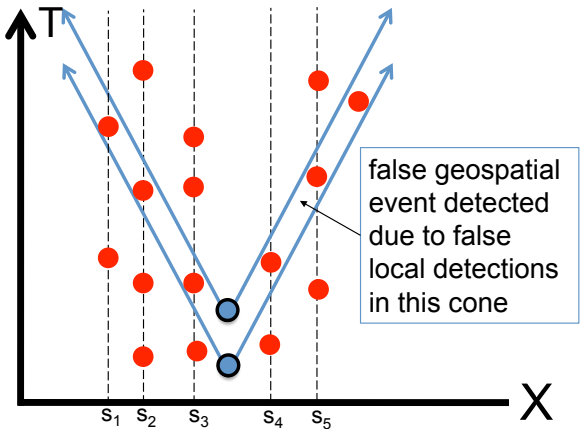
True positive and false negative. Intuitively from Fig. 1.5, a simple algorithm to detect geospatial events is as follows: detect a geospatial event when the number of sensors reporting events within a cone exceeds some multiple of the number of sensors that do not report events in that cone. All possible values of (x, t) can be the location of the apex unless there is prior knowledge on the event distribution. Fig. 1.5(a) shows two cones where the apex of each cone represents a true geospatial event. All the sensors generate sensor events in one of the cones, and no sensors generate events in the other cone. A geospatial event will be detected for the first cone — a true positive — but not for the second — which would be a false negative.

False positive. False positives can occur when false sensor detections happen to align by chance. Fig. 1.5(b) shows a false positive detection of a geospatial event because sensors generated detections due to local noise in a cone. The rate of false positives increases with uncertainty about the propagation of geospatial events, constraints on the speed of detection, and noise in sensors. Uncertainty about propagation speed is represented by cones with greater thickness, as shown in Fig. 1.5(c). Constraints on the speed of detection are represented by cones of shorter height, as the height of the cone represents the time from event initiation to detection; the number of sensors that can produce signals within a cone decreases with the height of the cone, and so detecting collections of true sensor events from false sensor events gets more difficult.

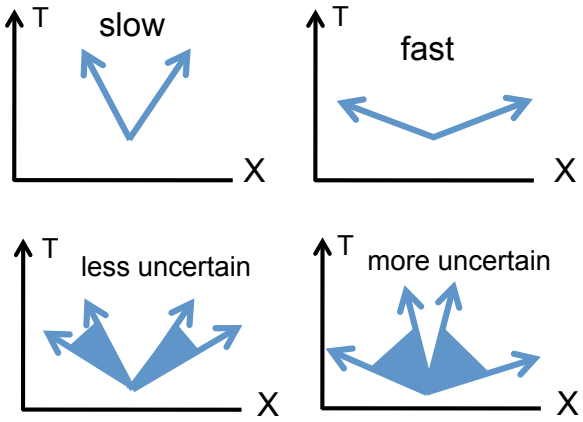
The definition of false positive depends on the application. For example, the U.S. Geological Survey reports many earthquakes that are too small to require a reaction. Warnings about such events would be considered to be false by most people, who only want to be alerted when they have to take action. How accurate the system can estimate these critical event parameters (e.g. location, magnitude) to avoid false warnings depends on many factors. For example, an event that propagates faster can be detected faster simply because the manifestation reaches more sensors in the same amount of time in comparison to a slower event. The parameters of a faster events, however, may not be estimated with the same accuracy as a slower event as shown in Fig. 1.5(c). These simple geometric intuitions will be further validated and quantified in Sec. 2.3 and Sec. 3.2.2.



(a) True positive and false negative resulting from a simple algorithm that aligns and counts the number of detections in a banded region



(b) False posoitve due to randomly aligned sensor false detections by chance



(c) Event speed and uncertainties affect the accuracy of detection and parameter estimation

Figure 1.5: Event detection model illustrated in 1D space-time plots, assuming an isotropic event model. The event is modeled to initiate at a single point (x_0, t_0). The blue lines show the propagation of event impact in space and time. The true sensor detections are marked as green dots, whereas the false sensor detections are red.

1.5 Example Applications

This thesis explores the commonality underlying widely different types of geospatial events. In particular, it studies two very different types of geospatial events in detail with the intent of both identifying concepts common to most geospatial events, and also understanding the limits of these general concepts. We study two problems in depth, rather than many problems shallowly, so that we can better understand the subtleties of the problem-specific issues of events that fall within the rubric of geospatial events. The two problems we explore are responding to earthquakes and radiation emission.

These two events can be considered as extreme cases of the geospatial event models introduced in Sec. 1.2.1. For example, the impact of radiation travels at the speed of light and is observed almost immediately after initiation, whereas the impact from earthquake travels at the speeds similar to that of the speed of sound. The same geospatial event detection model is used for both cases despite the fact that the speeds of propagation for light and sound are very different.

1.5.1 Radiation

The threat of terrorists with improvised nuclear devices, and nuclear reactor accidents such as Chernobyl and Fukushima have led to a great deal of research on radiation detection, including research on learning techniques for fast classification at ports [3].

Challenges. There are three major challenges for radiation detection with networked sensors.

1. Noise: The environment is naturally abundant in NORM (naturally occurring radiation active materials) sources, most of which are benign. Materials where NORM isotopes can be found include dirt, air, tiles, pavements, cat litter, and patients treated with radioactive therapies. NORM radiation can usually be distinguished from harmful ones (which emit gamma radiation) by examining the signature (spectrum) with high quality spectrometers. Unfortunately, the two types of radiation are indistinguishable with widely available inexpensive geiger counters that only record photon counts.

2. Occlusion: The environment has objects that absorb radiation. If the source of radiation is occluded from the sensors by objects, and the dimension and material of these objects are not factored into the calculation, the detection and parameter estimation results will be impaired.

3. Distance: The radiation signal drops off faster than the square of distance between the source and a sensor. This means detection is particularly difficult if no sensor is close to the source, i.e. if the network is sparse.

Previous work. The change in methods of terrorism and the emerging need of long term extensive monitoring of radiation level calls for a change in design. Early work in the distributed sensor network approach focuses on deploying sensors along streets. It has been found that with the same amount of cost, the networked sensing approach not only outperforms large portal sensors, but also shows improvements in combined signal-to-noise ratio and real-time detection [92] [48]. It has also been found that decentralized computation is important for persistent remote sensing [11].

Many techniques were developed for the detection and parameter estimation with a group of networked radiation sensors. The more popular ones are Sequential Probability Ratio Test (SPRT) and Maximum Likelihood Estimator (MLE). SPRT uses a adaptive thresholding technique that compares current measurements to previous baselines [49][74]. MLE method formulates the parameter estimation task as a multi-dimensional optimization problem. The problems of local minima and long search time with MLE are addressed by the mean-of-estimator method [76][21].

There exist several drawbacks to these classical methods, including discontinuous estimates, few results on convergence, and the fact that no prior information is used (e.g. the positions of known objects). Recent development of Bayesian methods address these issues [16], some supported by extensive realistic simulations [87][57]. A pure Bayesian approach suffers in computational efficiency when the parameters space is too large [58]. The problem is improved with a more efficient particle filter sampling approach [77], but the growth problem is still not completely resolved.

Despite the above mentioned problems with both classical and Bayesian approaches, these methods perform reasonably well with a small network of sensors (< 10). However, earlier work does not provide detailed analyses of system performance in terms of key network and event parameters, and without this information it is difficult to predict system performance as the network grows in size.

Analysis using the general geospatial model. How does the problem of radiation detection fit into the general models that were described? While a rigorous discussion will be provided later in the thesis, we first offer an intuitive discussion here.

Consider that a sensor detects a local anomaly by measuring a change in photon flux. Because photons strike a sensor in a Poisson manner, the time required for a sensor to detect the presence of a source is the time required to detect an increase in the long-term background flux due to the source. The time to detect an increase in flux depends on the amount of the increase. A smaller increase in flux can only be detected after a longer time whereas a large increase can be detected quickly. Because flux decreases as the inverse of the square of the distance from the source, the time to detect a change increases at least as fast as the square of the distance to the source. The representation in the simple 2-D model is shown in the Fig. 1.3(c).

1.5.2 Earthquake

Earthquakes are one of the the most devastating natural geospatial hazards in areas of frequent seismic activity. Because a reliable precursor of earthquake has not been found, the importance of robust earthquake early warning system (EEW) cannot be overstressed. The main purpose of an EEW system is to provide realtime event estimates to the distant site before the strong shacking (S-wave) arrives.

Challenges. Earthquake waves propagate in a very complex manner that depends on the depth of the quake, the event magnitude, and earth structure. An important part in EEW application is to determine the first P-wave arrival at each sensor; however, exact determination is difficult when multiple aftershocks or multiple independent quakes are occurring simultaneously and the sensors observe dependent and interfered measurements. The problem is even more difficult with less precise and noisy sensors.

Previous work. There is a rich body of work on EEW applications regarding earthquakes. An example is the Japanese seismic network, which consists of roughly 1,000 high precision sensors throughout inland Japan (about 1 station every 400 km^2). The current early warning system running on the network uses a combination of classical methods with single or multiple station measurements to rapidly estimate the location and magnitude of an earthquake after first local detection by a station [43]. A second example is the California Integrated Seismic Network (CISN) that consists of about 350 sensors from Southern California Seismic Network (SCSN)[5] and 412 sensors from Northern California Seismic Network (NCSN)[6]. The EEW system running with CISN also includes several classical methods. One of these is the $\tau_c - P_d$ approach, which estimates the S-wave magnitude and peak ground velocity (PGV) with the P-wave pulse width (τ_c) and the peak initial 3-second displacement amplitude (P_d) [88]. This method has been proven successful for real-time estimation in a series of events [9].

In addition to classical methods, a Bayesian approach for EEW was introduced by Cua and Heaton in 2005. The *Virtual Seismologist* project constructs the likelihood function based on six channels of measured ground motion (maximum vertical acceleration, maximum horizontal accelerations (2-axes), velocity, and displacement) [26][24]. Prior information, such as the identified fault lines, can be easily incorporated into the calculation. A simple grid search method is used to compute the posterior distribution. Another approach includes the development of *PreSEIS* by Bosë *et al.* that uses a 2-layer feedforward neural network learned on historical data to estimate realtime event parameters, including hypocenter location, moment magnitude, and the expansion of evolving earthquake rupture [10]. PreSEIS has the advantage over classical methods in that they can han-

[5] http://www.scsn.org/seisstations.html
[6] `http://www.ncedc.org/ncsn/`

dle nonlinear and noisy data that is not well explained with existing models; however, as with all learning approaches the performance is limited by the completeness of training data set.

All the previous work on EEW applications assumes constant access to accurate measurements from each sensor (clean 3-axes acceleration or velocity). However, with lower quality sensors (to meet cost constraint and to increase coverage), the sensor measurements are unreliable due to electronic or environment noise. For example, the sensor orientation may not be consistent through the network, and instead of 3-axis accelerations, we may need to resort to use a single combined acceleration measurement. The data can also be intermittent, depending on the cell or wireless network availability. With these constraints, it is important to establish new models and develop new algorithms that are scalable and noise-tolerant.

Participatory sensing. The intensity of shaking at a location during a seismic event depends not only on the distance to the hypercenter, but also the local earth and building structures that are difficult to capture in a generic model. A dense network allows for creating a high resolution intensity map of shaking ("shake map") that helps first responders assess the priority of emergency relief. The data also helps scientists better understand the composition of substructures. An example of such dense network is the temporary Long Beach dense network (Fig. 1.1), which consists of ∼5,000 sensors in 35 km^2 [7]. During the six months of data collection, more than 200 small events (<M2.0) were detected. This number is far greater than that reported by USGS.

Unfortunately, the long-term deployment of a dense network is infeasible due to the high cost of traditional seismometers (∼$10,000). As a comparison, SCSN would need another 20,908,935 stations to reach the same density as the Long Beach network. As a result, alternative sensing methods were developed. *Did you feel it* employs volunteers to rate the shaking they experience during an event, and with the inputs, it is able to create surprisingly accurate shake maps [7]. Fig. 1.6 is an example of a shake map created for the 1994 M6.7 Northridge earthquake based from 2208 volunteer inputs[8].

Other data collection methods exploit the wide availability of inexpensive accelerometers in modern day consumer electronics. The *iShake* project uses smartphones as sensors for seismic activity monitoring [33]. The *Community Seismic Network* includes standalone USB accelerometers ("Phidget" sensor) in addition to smart phone sensors ("Android" sensor) [22]; the joint network has reached over 200 sensors distributed across the greater Los Angeles area, and has been reliably detecting all events of M2.0 and above reliably in Southern California since 2011.

Analysis using the general geospatial model. This thesis covers both high quality and low quality sensor networks for earthquake detection. Although the sensor noise profiles are different, in

[7] `http://www.gps.caltech.edu/~clay/EQmovies/EQmovies.html`
[8] Figure source:`http://pubs.usgs.gov/fs/fs030-01/`

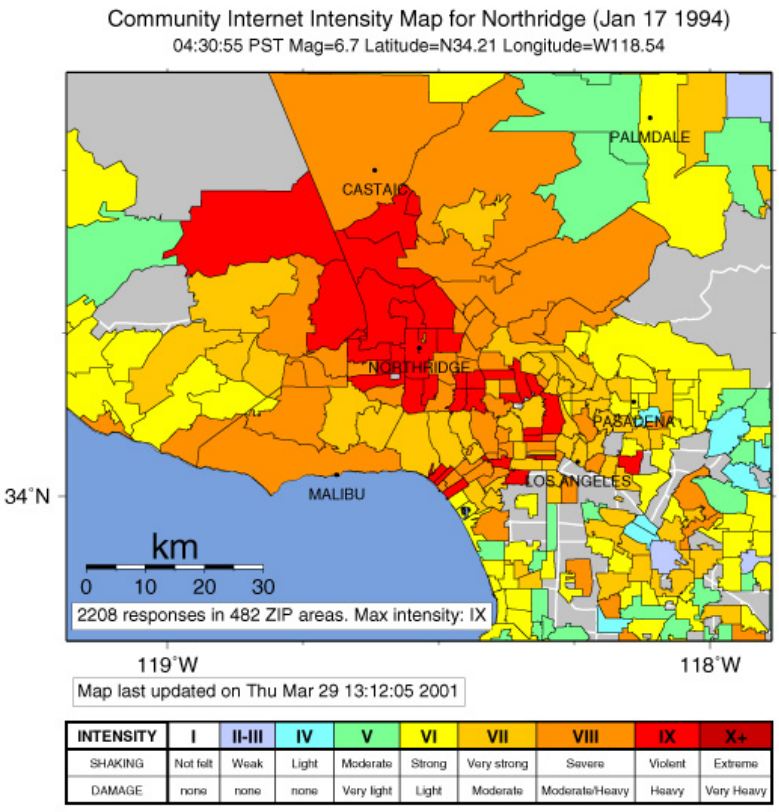


Figure 1.6: *Did you feel it* shake map created for the M6.7 Northridge earthquake on January 7, 1994 with 2208 independent inputs across 482 zip codes.

both cases, the P- and S-wave takes some finite amount of time to reach the network. In most of the analyses, we assume a constant event propagation model such as the one illustrated in Fig. 1.3(a). In the few cases where a more sophisticated speed model is available, a variable speed model, as shown in Fig. 1.3(c), is used instead.

1.6 Chapter Organization

This thesis is organized to address the three key requirements of a system for event detection — *speed*, *accuracy*, and *scalability*.

Speed. Determining whether an event has occurred is often the first step in event detection. Rapid detection is of tremendous importance in some applications. For example, in earthquake early warning systems, a few seconds of warning can prevent massive loss in lives and capital. However, improvement in detection speed is directly related to increases in the number/quality of sensors and/or sacrifices in detection accuracy. This tradeoff of system performance in terms of key network and event parameters is studied in Chapter 2.

Accuracy. From the time-series network observations, event parameters can be estimated. These estimates have direct impact on choosing the right response to an event. For example, if the detected radiation source is identified as benign, the system should not issue a warning. On the contrast, if an area is expected to experience strong shaking based on the estimated intensity and hypocenter of an earthquake, then a timely warning is crucial. The problem of parameter estimation is difficult in the presence of noise, and is further complicated by unconstructed parameter space, traveling event source, multiple event sources, and mobile sensors. Chapter 3, Chapter 4, and Chapter 5 discuss how the quality of estimates can be improved under these complications.

Scalability. With the advancement of mobile technology and the decrease in price of sensing devices, building a large scale dense sensor network is no longer impossible; however, such large noisy networks call for updates in system design, both in terms of software architecture and algorithm development. For example, it is infeasible to stream all the raw measurements to a centralized fusion center due to the large volume of data, and the aggregation algorithms must take into account the variability in sensor quality, installation method, and regional noise. Some of these issues are addressed in Chapter 6 in terms of rapid detection.

Chapter 2

Detection Time and Confidence Tradeoffs

At any given instant during the unfolding of an event, accumulated data since the initiation of the event to the current instant can be observed; a goal of the system is to use this accumulated information to predict the future evolution of the event. As time progresses, the amount of observed information about the event generally increases. One may thus reasonably expect that predictions will become more accurate over time; however, faster response provides more time for people and systems to respond, and as a result allows damages to be confined and minimized. Many geospatial problems deal with the question: given observations up to a point in time, should systems initiate responses or should they wait for more corroborating information?

Detection time and confidence tradeoffs. Several parameters play a role in the tradeoff between detection time and confidence. Some of these parameters, such as environmental noise and the event's location and intensity, are not controllable. Parameters that can be controlled include the number, quality, variety, and location of sensors. The optimal selection of these parameter values under the constraint of limited resource (e.g. power, bandwidth) is a difficult non-convex optimization problem.

Receiver Operating Characteristic curve. The key metric used in this chapter and throughout the rest of the thesis is the Receiver Operating Characteristic (ROC) curve. ROC curve plots the probability of true positives as a function of the probability of false positives. By tracing the points on the curve, one can tradeoff true positive with false positive.

Chapter organization and contributions. This chapter presents the first thorough mathematical analyses and Monte Carlo methods to help understand the fundamental tradeoffs between detection time, the probabilities of false positives and true positives, system cost, numbers of sensors, benefits of data fusion, and communication load. We begin with the analysis on the asymptotic characterization of detecting a geospatial event with sensor networks, under the basis of no central-

ized computation (and therefore no communication cost). This analysis serves as a lower bound on the detection performance (Sec. 2.1). Better detection performance can be achieved by combining realtime measurements from multiple sensors at a fusion center, though doing so increases the communication overhead. Sec. 2.2 presents the first rigorous quantification of data fusion benefit in terms of reduction in detection time. This analysis leads to a new detection bound for general geospatial event detection with cooperating sensors, a detection bound that relates detection confidence to key network parameters (Sec. 2.3).

The focus of this chapter is restricted to analyzing the problem of detection, i.e. determining whether an event has occurred. The problem of estimating event related parameters, e.g. localization, will be presented in later chapters.

2.1 Asymptotic Characterization

We first determine what a single sensor can detect and later consider what can be detected by collections of cooperating sensors. We begin with a simple sensor that measures the raw manifestation of the event. The system issues an alert exactly when the measurements collected in time t by the (single) sensor exceeds some threshold value τ. The greater the value of τ, the lower the probability of false positives, but also the lower the probability of true positives. By varying τ we can study the tradeoff between the probabilities of false positives and true positives.

2.1.1 Definitions

We begin by comparing probabilities of the following two alternatives

1. No event has occurred in time T. The sensor observes only background noise of strength Γ.
2. An event of magnitude μ at distance r from the sensor has occurred in time T. The sensor observes the event manifestation $f(\mu, r, T)$, in addition to the background noise Γ.

If the system falsely declares detection in the first scenario, then we say there is a *false positive*. If the system correctly detects in the second scenario, then it is a *true positive*. The strength of the signal from the event depends on all the parameters mentioned, as well as the quality of the sensor that is captured by the sensor detection function ϕ, i.e. sensor reports measurements given by $\phi(f(\mu, r, T))$.

In our radiation example, event manifestation function f is the *Poisson* probability density function where the number of events k (photon registered) depends on source strength μ, distance to source r, observation time T, sensor sensitivity and dimension A, and the photon absorption rate α.

$$f(\mu, r, T) = poisson(k; \lambda, T) = \frac{(\lambda T)^k e^{-(\lambda T)}}{k!}$$

where λ is the effective source strength measure at the sensor,

$$\lambda = \frac{A\mu e^{-\alpha r}}{r^2} \tag{2.1}$$

Let pFP be the probability of a false positive; that is the probability of observing τ or more photons from the background of intensity Γ in time t:

$$pFP = \sum_{k \geq \tau} poisson(k; \Gamma, T) \tag{2.2}$$

Let pTP be the probability of a true positive; that is the probability of observing τ or more photons from the combination of background and the source in time t:

$$pTP = \sum_{k \geq \tau} poisson(k; \Gamma + \lambda, T) \tag{2.3}$$

2.1.2 Asymptotic Analysis

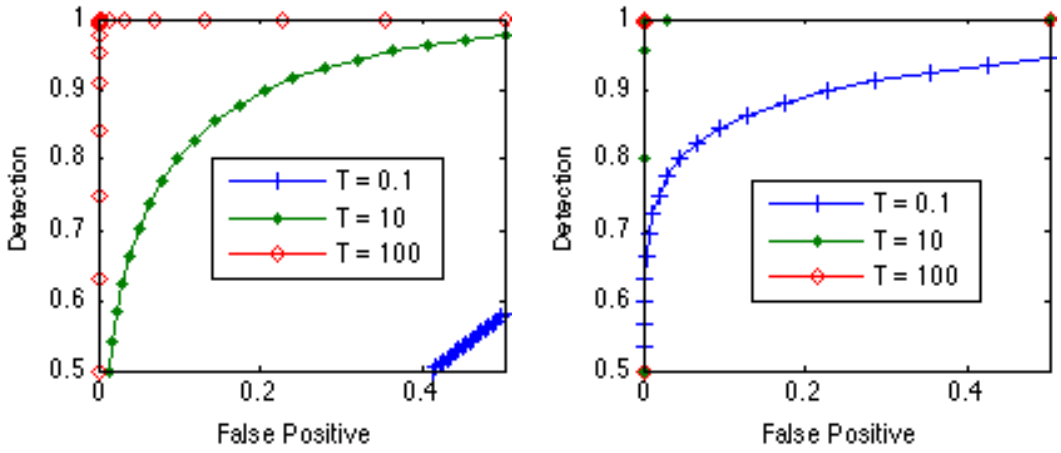
Intuition. We begin the analysis with some plots that provide insight to the problem. Fig. 2.1 are two ROC curves that plot Eq. (2.2) as a function of Eq. (2.3) by varying the threshold τ for two fixed sets of T, Γ and λ. It is clear that the ROC curves improve (higher pTP for a fixed pFP) with increasing observation time T — the time at which the system can use to decide whether a source is present or not. The curves also show that moving the source further away from the sensor or increasing the background intensity reduces the quality of the ROC curve. By selecting a fixed pFP, we can re-plot Fig. 2.1 as a function of distance to the source r to reach a desired pTP, as shown in Fig. 2.2. The figure shows the quadratic growth in detection time as the distance increases.

Asymptotic behavior. Next, we rigorously define the asymptotic relationships between pTP, pFP, T, λ, and Γ; these asymptotic relationships will give insight into how to design detection systems.

A Poisson distribution can be approximated, with error low enough for our purposes, by a normal distribution when the count of events $k \geq 10$. Let μ_s and σ_s be the mean and standard deviation of aggregated observations in time T when a source is present. Let μ_0 and σ_0 be the corresponding values when the source is absent. We then have

$$\mu_s = (\lambda + \Gamma)T, \quad \sigma_s = \sqrt{(\lambda + \Gamma)T}$$
$$\mu_0 = \Gamma T, \quad \sigma_0 = \sqrt{\Gamma T}$$

Let bFP be the upper bound on the probability of false positives pFP. Let Z_{FP} be a positive



(a) $\lambda = 2$, $\Gamma = 8$ (b) $\lambda = 32$, $\Gamma = 8$

Figure 2.1: ROC curves of the probability of detection.

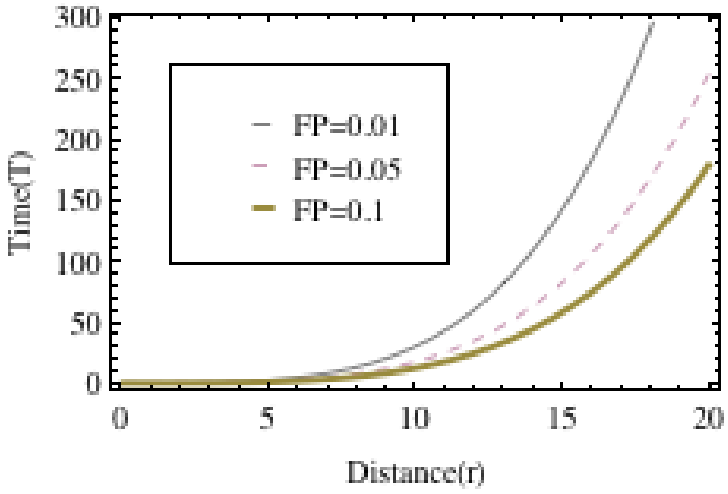


Figure 2.2: Time to detect as a function of distance. $\Gamma = 8$, $\mu = 200$, $pTP = 0.99$.

real value such that the probability of values in a standard normal distribution exceeding Z_{FP} is bFP. Then

$$\tau - \mu_0 = \tau - \Gamma T \geq Z_{FP}\, \sigma_0$$

For example, when $bFP = 0.01$ then $Z_{FP} = 2.33$. Similarly, let bFN be the upper bound on the probability of false negatives (i.e. 1-pTP), and let the probability of values in a standard normal distribution being less than bFN be Z_{FN}. Then

$$\mu_s - \tau = (\lambda + \Gamma)T - \tau \geq Z_{FN}\, \sigma_s$$

Consider the point on the ROC curve where $Z_{FN} = Z_{FP}$ (i.e. 1-pTP = pFP), and let us call this

value z. From the above equations we get

$$z = \frac{\lambda\sqrt{T}}{\sqrt{\Gamma} + \sqrt{\lambda + \Gamma}}$$

For the impact intensity λ much lower than the background intensity Γ:

$$\lambda \ll \Gamma : \quad z \approx \frac{\lambda\sqrt{T}}{2\sqrt{\Gamma}} \tag{2.4}$$

and for impact intensity λ much greater than the background intensity Γ:

$$\lambda \gg \Gamma : \quad z \approx \sqrt{\lambda T} \tag{2.5}$$

Eq. (2.5) represents the case when the signal strength far exceeds noise. In such case, the detection problem is trivial. We focus on the case when the signal strength is much weaker than noise as shown in Eq. (2.4), and study how the network performance changes as the key parameters change.

2.1.3 Interpretation

Substituting effective source strength at the sensor λ from Eq. (2.1) into Eq. (2.4) and rearranging, we get the following relationship with respect to observation time T

$$\frac{\mu}{\Gamma} \ll \frac{e^{\alpha r} r^2}{A} : \quad \Rightarrow \quad T \approx \frac{4 z^2 \Gamma r^4 e^{2\alpha r}}{A^2 \mu^2} \tag{2.6}$$

We use this equation to analyze the following situations.

1. **Increasing distance between sensors**: When r is large, the time T to make a decision increases more rapidly than the fourth power of r. For example, with $A = 200$, $K = 3.2$, and $r = 20$ meters, then a 25% increase in r requires a 150% increase in T. Thus timely detection forces dense deployments of sensors.

2. **Reducing sensor sensitivity**: If the sensor sensitivity A is halved and everything else remains unchanged, then the time T to detection is quadrupled.

3. **Increasing background noise**: If the background intensity Γ is doubled, then the time to make a decision is doubled as well, if other parameters remain unchanged; if we want to make decisions just as quickly then A must increase by $\sqrt{2}$ or 1.4.

4. **Increasing sensor density**: Let D be the sensor density: it is the surveillance area divided by the number of sensors. To a first approximation, sensor density is proportional to $1/r^2$

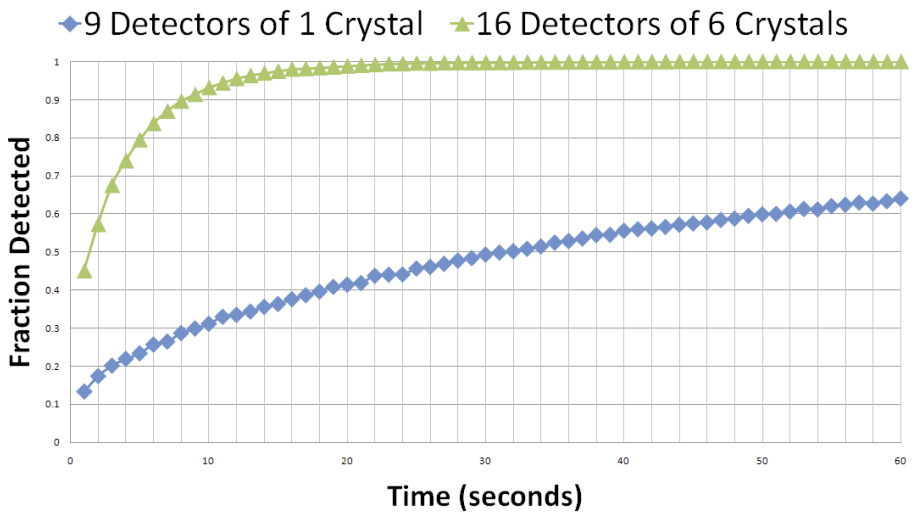


Figure 2.3: Increasing the density of sensors improves detection. The graph compares detection efficiency using nine sensors with that obtained using 16 sensors that are each 6x more sensitive, under otherwise identical conditions.

where r is the average spacing between sensors laid in a grid. For large r we see that T *increases more rapidly than* Γ/D^2A^2.

These analyses show that a dense network is required to rapidly detect a weak event. It should be noted that with such a dense network, the probability that some sensors are near the event source is high. This scenario is described by Eq. (2.5) as opposed to Eq. (2.4), and the detection problem becomes much easier since observation time T is affected less by background noise Γ and sensor sensitivity A.

$$\frac{\mu}{\Gamma} \gg \frac{e^{\alpha r} r^2}{A} : \quad \Rightarrow \quad T \approx \frac{z^2 r^2 e^{\alpha r}}{A\mu} \tag{2.7}$$

Simulation results in Fig. 2.3 show how these parameters affect detection performance. A sparse network (blue curve) can only detect with 0.3 confidence in $T = 10$ seconds, whereas a dense network of roughly twice the sensor density and 6x times more sensitive (green curve) can detect with 0.92 confidence in the same amount of time. While this analysis focuses on the application of radiation, the principles can be applied to a wide array of geospatial events as most events have similar underlying behaviors. As the analysis suggests, inexpensive (noisier and less sensitive) sensors can achieve the same performance as that of high quality ones by increasing the density. We will further explore this idea in Chapter 6.

2.2 Combining Sensor Data

The asymptotic analysis presented in Sec. 2.1 is based on the assumption that all sensors are homogeneous and operate independently, i.e. there is no communication between sensors. In reality,

single sensor detection performance is inhomogeneous and varies as a function of distance to the event initiation. The network detection performance can be improved by combining real-time measurements from sensors, although doing so increases the infrastructure cost as well as communication cost, because it now requires data to be sent from the sensors to a centralized fusion center. Understanding the performance gain of data fusion is therefore critical in designing a real system for event detection. This section quantifies the benefit of data fusion in terms of the reduction in detection time.

2.2.1 Classical Statistics Analysis

We consider the problem of correctly detecting an event when one is present with a probability of at least bTP, and falsely detecting an event when it is absent with probability of at most bFP. In other words, bTP is the lower bound on the probability of a true positive pTP, and bFP is the upper bound on the probability of a false positive pFP. We present analytical analyses for two communication models: (a) when there is no communication between sensors and decision is made independently (no fusion), and (b) when sensors communicate and a decision is made on the combined measurements (fusion). Results from the two cases are then compared to get a quantified measure of fusion benefit.

Preliminaries. The analyses use the following assumptions:

1. All sensors observe the same amount of time unvarying background noise Γ.
2. The event manifestation (photon emission) is isotropic.
3. There exists no obstacle between the event initiation (source) and the sensors.
4. The field either contains no or exactly one event (radiation source).
5. The network makes the decision on whether an event has occurred after observing for time T.

Detection test statistics. We adopt a simple threshold test statistics such that the null hypothesis $\mathcal{H}_0$ is rejected if and only if some function of the sensors' combined measurements exceeds a threshold. Different functions and thresholds are used according to whether decisions are being made with or without data fusion.

- No fusion: The null hypothesis is rejected if the measurement (photon count) from any sensor exceeds $(\Gamma + K\sigma_{null})$, where K is a constant determined by the bounds on the probabilities of true positives and false positives, bTP and bFP, with respectively.

- With data fusion: The sensor measurement can be combined in many ways. We study one of the approaches that the null hypothesis $\mathcal{H}_0$ is rejected if and only if any of the following conditions described hold:

    1. The measurement from any sensor exceeds $(\Gamma + K_1\sigma_{null})$, where K_1 is a constant. This corresponds to the no-fusion case, except that the threshold K_1 is different from (and is larger than) K. If the source is near a corner of the square, then the photon count for the sensor at that corner is likely to exceed this threshold. If, however, the source is far away from every corner, then this threshold is unlikely to be exceeded.
    2. The average measurements of any pair of neighboring sensors exceeds $(\Gamma + K_2\sigma_{null})$, where K_2 is a constant. The reason for fusing data from adjacent pairs of sensors is to deal with the possibility of a source being along an edge of the square, and near the middle of the edge. For example, if the length of the square is 20 meters, and the source is midway along an edge of the square then r^2 is 100 for the two nearest sensors and is 500 for the two farthest sensors, where r is the distance between the source and sensor.
    3. The average measurements of any group of four neighboring sensors exceeds $(\Gamma + K_3\sigma_{null})$ where K_3 is a constant. This threshold deals with the possibility of a source near the center of any square.

This design of detection statistics eventually leads to the development of the `K-sigma` algorithm that will be introduced in Sec. 3.1.1. The values of K_1, K_2 and K_3 are determined to satisfy the given bounds on the true positive and false positive probabilities. Note that by setting K_1 to K, and K_2 and K_3 to infinity, the fusion case becomes identical to the no fusion case. As a result, we can ensure that the fusion algorithm is at least as good as the independent algorithm; we will, however, select the values of K_i so that the fusion algorithm can potentially outperform the independent algorithm.

2.2.2 Quantifying Fusion Benefit

Metric. Before we can compare the detection performance for the two cases — fusion and no fusion, we need to establish a metric for comparison. An ideal metric is time. More precisely, it is the *time to detect* an event since initiation under the constraint that the system false positive rate is bounded above, $pFP \leq bFP$, and the system true positive rate is bounded below, $pTP \geq bTP$.

Two sensors. Let us first consider a simple scenario with two sensors at a distance R apart. Let $G(\mu, \sigma; n)$ and $F(\mu, \sigma; n)$ be the cumulative distribution and probability density functions for a Poisson random variable k with parameter μ; $G(\mu, \sigma; n)$ is the probability of $k \leq n$, and $F(\mu, \sigma; n)$

is the probability of $k = n$. Let Γ be the expected contribution from the background at a sensor in time T. Because there are only two sensors, the test statistics described in Sec. 2.2.1 is modified. Let k_1 and k_2 be the measurements observed in time T from sensor s_1 and s_2, and we say there is a detection when

1. No fusion: the measurement (photon count) of any sensor exceeds a threshold $Q = \Gamma + K\sigma_{null}$, i.e. $(k_1 \geq Q) \vee (k_2 \geq Q)$

2. With data fusion: the measurement of any sensor exceeds a threshold Q_1 or the sum of the counts of the two sensors exceeds a threshold Q_2, i.e. $(k_1 \geq Q_1) \vee (k_2 \geq Q_1) \vee (k_1 + k_2 \geq Q_2)$.

Recall from Sec. 2.1 that the expected and standard deviation of sensor measurements when an event is absent (μ_0 and σ_0) and when an event is present (μ_s and σ_s) are:

$$\mu_s = (\lambda + \Gamma)T, \quad \sigma_s = \sqrt{(\lambda + \Gamma)T}$$
$$\mu_0 = \Gamma T, \quad \sigma_0 = \sqrt{\Gamma T}$$

From these, we can define the probability of false positive pFP and true positive pTP as a function of threshold Q_i's for the fusion and no fusion cases.

1. No fusion: A detection is made when $(k_1 \geq Q) \vee (k_2 \geq Q)$, therefore

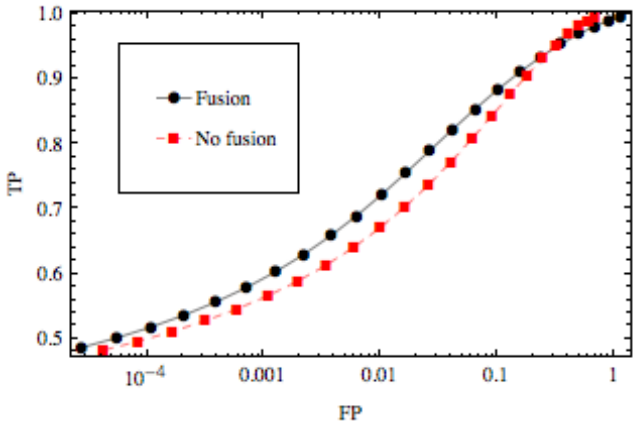
$$pFP = 1 - G(\mu_0, \sigma_0; Q)^2$$
$$pTP = 1 - G(\mu_{s1}, \sigma_{s1}; Q)G(\mu_{s2}, \sigma_{s2}; Q) \tag{2.8}$$

   where μ_{s1} and σ_{s1} are the mean and standard deviation of the measurements observed at sensor s_1. $\mu_{s1} = (\lambda_{s1} + \Gamma)T$ and $\sigma_{s1} = \sqrt{(\lambda_{s1} + \Gamma)T}$. The same definition applies to μ_{s2} and σ_{s2}.

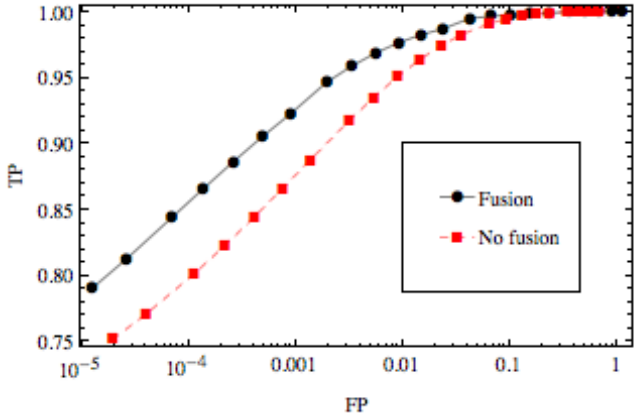
2. Fusion case: A detection is made when $(k_1 \geq Q_1) \vee (k_2 \geq Q_1) \vee (k_1 + k_2 \geq Q_2)$. There are three scenarios when this condition holds: (a) $k_1 \geq Q_1$, or (b) $(k_1 < Q_1) \wedge (k_2 \geq \min(Q1, Q_2 - k_1))$. Since the two disjuncts are mutually exclusive their probabilities are summed:

$$pFP = (1 - G(\mu_0, \sigma_0; Q_1)) + \int_0^{Q_1} F(\mu_0, \sigma_0; x_1)\,(1 - G(\mu_0; \min[Q_1, Q_2 - x_1])\ \mathrm{d}x_1$$
$$pTP = (1 - G(\mu_{s1}, \sigma_{s1}; Q_1)) + \int_0^{Q_1} F(\mu_{s1}, \sigma_{s1}; x_1)\,(1 - G(\mu_{s2}, \sigma_{s2}; \min[Q_1, Q_2 - x_1])\ \mathrm{d}x_1$$

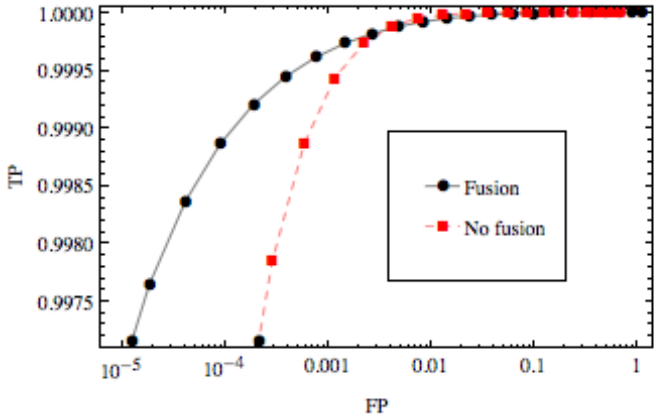
Assuming that a source is placed anywhere on the straight line between sensor s_1 and s_2 with uniform probability, by varying Q_i's, we can construct ROC curves for different detection time T. Fig. 2.4 shows the ROC curves for no fusion and fusion algorithms at $T = [3, 15, 50]$ seconds. The



(a) $T = 3$ second



(b) $T = 15$ second



(c) $T = 50$ second

Figure 2.4: ROC curves with and without data fusion. $\Gamma = 8T$, $\Lambda = 200T$, $R = 20$. The detection performance with data fusion clearly outperforms the one without data fusion, especially in the lower pFP regime.

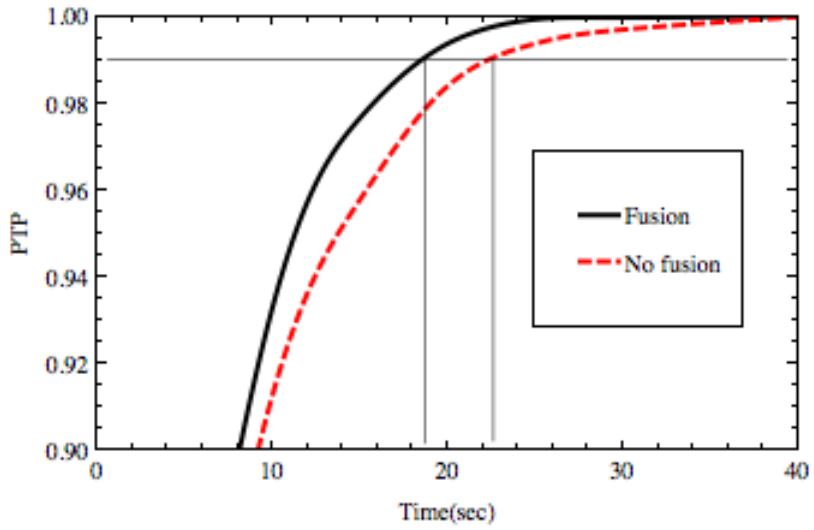


Figure 2.5: pTP as a function of time with a fixed $pFP = 0.01$ using the optimal Q_1 and Q_2. The amount of fusion benefit can be quantified as 4 seconds if the desired detection confidence is 0.99.

values of the thresholds with the fusion case, Q_1 and Q_2 can be optimized for each desired pFP and pTP combination; however, these plots do not explore all possible combinations of $[Q_1, Q_2]$. The suboptimal choice accounts for the crossover on the upper right corner in Fig. 2.4(a).

Fig. 2.4 shows that for all T, detection with fusion indeed outperforms the case with no fusion, especially in the low false positive regime. Furthermore, these graphs can be replotted as a function of detection time T by fixing a desired false positive value. The performance of no fusion and fusion can then be measured in terms of the *time required* to reach a certain detection confidence (true positive rate). An example is shown in Fig. 2.5, which shows that it takes ~ 19 seconds and ~ 23 seconds to reach $pTP = 0.99$ for the case of fusion and no fusion, respectively. In other words, the benefit of data fusion under the constraints $[pFP, pTP] = [0.01, 0.99]$ can be quantified as 4 seconds, i.e. we can detect 4 seconds faster if data can be combined at a centralized center.

Four Sensors. Extending the analysis from two sensors, we now consider the arrangement of four sensors placed at each corner of a square field. An event (radiation source) can occur anywhere with uniform probability inside the field. Although similar studies as the ones done with two sensors can be carried out here, we resort to Monte Carlo simulation as the complexity of the exact analyses grow exponentially with the number of sensors. Fig. 2.6 shows the detection capability of a four sensors ensemble, with and without fusion, as a function of the source position. The vertical axis is the logarithm of the detection variable K; a larger K implies that the event can be more easily detected. As Fig. 2.6(a) and Fig. 2.6(b) suggest, an event is most difficult to detect when it is along the edges and in the middle of the field where the combined signal-to-noise ratio is the lowest. The increase in K along the edges in Fig. 2.6(b) demonstrates that by combining data from sensors,

we can effectively reduce the detection time in an otherwise difficult to detect area with a sparse network. In this particular setup, it gives approximately 10% improvement in detection time.

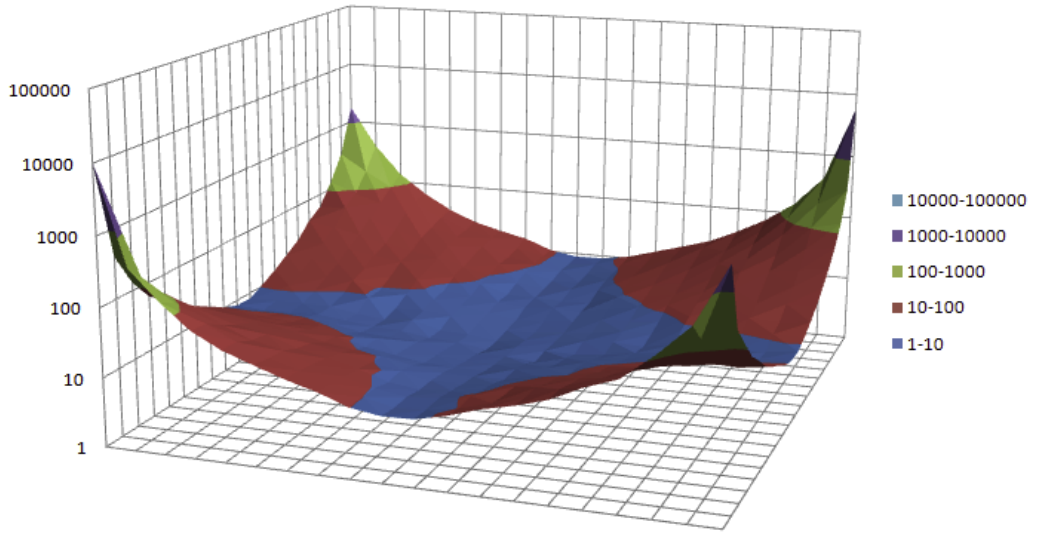
2.3 General Detection Bound

In Sec. 2.1 and Sec. 2.2, we study the performance tradeoffs of detection in sensor networks, assuming raw floating point sensor observations of event impact are available. This corresponds to the *streaming* communication model discussed in Sec. 1.3.1. This section extends the discussion from the streaming model to a *compressed* model, such that the network makes decisions based on a set of much compressed information, e.g. binary bits from sensors that indicate whether something abnormal has been observed. We prove the detection bound of this model for a simple scenario.

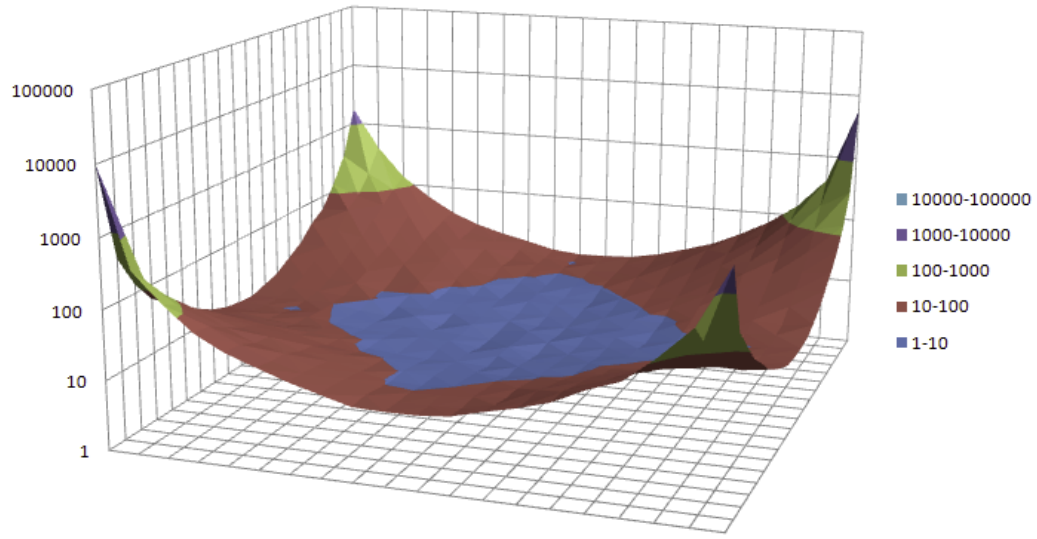
2.3.1 Preliminaries

We assume a binary sensor detection model. The sensor observes the environment continuously but only communicates a one-bit information together with a timestamp when an abnormal measurement has been observed. The analysis is based on the following assumptions.

1. The event can be modeled as a point source with origin location x_0 and start time t_0
2. The event propagates through the network at a constant speed v.
3. The sensor response is homogeneous when there is no event, i.e. all sensors experience the same amount of noise.
4. The sensor response is inhomogeneous when there is an event, i.e. the probability of detection varies as a function of distance to the event.
5. Each sensor has an internal clock that may not be perfectly synchronized. The timing error (difference between sensor's clock and a global clock) can be modeled as a Gaussian random variable $\mathcal{N}(0;\sigma)$.
6. The prior probability distribution is uniform, i.e. the event can start anywhere and anytime in space-time.
7. All n sensors in the network are semi-randomly distributed (e.g. they do not all lie on a single point or single line). The overall density d is uniform to a first approximation.
8. The event always occurs in the center of the network. Note that this is the optimal situation, as the amount of observable signals is maximized. As a result, the analysis serves as an upper bound on the system performance.



(a) No fusion



(b) With fusion

Figure 2.6: Event detection capability for four sensors as a function of event location. The vertical axis is the detection test statistics — larger value implies that it is easier to detect at that location. Data fusion improves the detection time at locations along the edges of sensors and in the middle of the field.

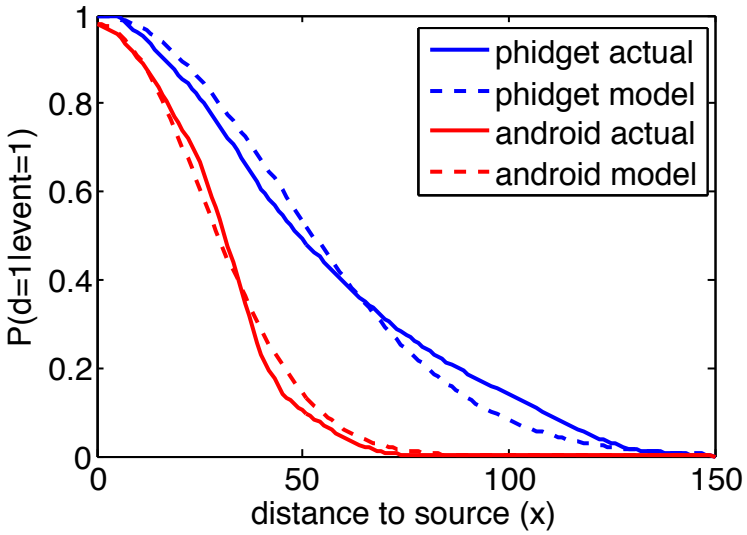


Figure 2.7: Comparison of parametric detection attenuation model to the actual data collected for two inexpensive vibrational sensors of different quality. The curves are constructed for the false positive constraint of 0.01. The detection performance of the lower quality *Android* sensor drops faster than that of *Phidget*. For both sensors, the parametric model closely captures the shape of the attenuation.

2.3.2 Sensor Model

The sensor response under the binary detection model can be denoted as a binary random variable X. X takes the value 1 (i.e. detects an anomaly) with different probabilities depending on whether an event is present. Because the impact of event manifestation decreases as a function f of distance to the event origin, the probability of detection when an event is present also varies for each sensor as a function of distance.

Let $\phi(H)$ denotes the sensor true positive probability $\mathbb{P}[X = 1|E = 1]$ under a desired false positive constraint for event H. Because the sensor sensitivity varies for different sensor type, $\phi(H)$ also varies for different sensors and can be parametrized as a function of sensor and event quality measure measures α and β, and distance to the event origin r.

$$\phi(H) = \alpha \exp(-\beta r^2) \tag{2.9}$$

Fig. 2.7 compares this parametric model to experimental results for two types of vibrational sensors of different quality — *Phidget* and *Android*. As expected, the detection probability $\phi(H)$ decreases with distance r. In addition, a faster decrease is observed with the lower quality *Android* sensor (the built-in 12-bit accelerometer in smart phones) than that with the higher quality *Phidget* sensor (16-bit accelerometer). In both cases, the parametric model with properly selected α and β closely captures the shape of probability attenuation.

2.3.3 True Positive Bound

We present the following bound on the system detection performance.

Theorem 1. *Under the aforementioned assumptions, and given a maximum allowable system wide false positive rate of g, $0 \leq g \leq 1$, the system's true detection rate is bounded by the density of the sensor distribution d, the length of the aggregation window time t, the speed of the event manifestation v, and the uniform noise rate q*

$$pTP \geq 1 - \exp\left(-2\left(\frac{\pi\alpha d}{\beta}\left(1 - e^{-\beta v^2 t^2}\right) - \left(\pi v^2 t^2 dq + vt\sqrt{\frac{-\ln(g)d\pi}{2}}\right)\right)^2 / \pi v^2 t^2 d\right) \tag{2.10}$$

Proof of Theorem 1. We begin the proof with analyzing pTP and pFP, given a simple detection test statistic.

Simple test statistic. The system declares detection when the number of sensor detections exceeds a certain threshold m within time t in an area of size $\pi R^2, R = vt$, where v is the speed of the event manifestation. This test statistic is equivalent to selecting an aggregation time window t and aggregating all sensor detections in regions the event may have affected. As shown Sec. 2.2, the exact analysis on detection performance is grossly complicated by the origin location of the event. Here we focus on the optimal scenario, where the event origin is always in the center of the network to get an upper bound on the system performance.

Probability of true positive. Recall that the sensors are distributed uniformly at a density d. Let K be the random variable that indicates the number of sensor detections. The probability that exactly k sensors detect the event in t is

$$\mathbb{P}\left[K = k | E = 1\right] = \sum_{A \in F_k} \prod_{i \in A} p_i \prod_{j \in A^c} (1 - p_j) \tag{2.11}$$

where F_k is all the subset of size k of the set $\{1, 2, \cdots, n\}$, $n = \pi R^2 d$. A^c is the complement of set A. p is the sensor detection probability, $p = \mathbb{P}\left[d = 1 | E = 1\right]$. Recall that p_i depends on the distance r_i to the event origin. The expected value of K can be computed as

$$\mathbb{E}\left[K | E = 1\right] = \sum_{k=1}^{n} k\, \mathbb{P}\left[K = k | E = 1\right] = \sum_{k=1}^{n} k \sum_{A \in F_k} \prod_{i \in A} p_i \prod_{j \in A^c} (1 - p_j) \tag{2.12}$$

Observe that Eq. (2.11) is a*Poisson Binomial Distribution.* Eq. (2.12) is then the mean of distribution and can be simplified as the sum of p_i for all sensor s_i. Given the parametrized model

of p_i as in Eq. (2.9), the expectation becomes

$$\mathbb{E}\left[K|E=1\right] = \sum_{i}^{n} p_i = \int_{r=0}^{R} (2\pi r d)\ \left(\alpha e^{-\beta r^2}\right) \mathrm{d}r = \frac{\pi\alpha d}{\beta}\left(1 - e^{-\beta R^2}\right)$$

The network declares a detection when $K > m$. From this test statistic we can compute the system true positive rate as:

$$\begin{aligned} pTP &= \mathbb{P}\left[K \geq m|E=1\right] \\ &= \sum_{K=m}^{n} \sum_{A\in F_K} \prod_{i\in A} p_i \prod_{j\in A^c} (1-p_j) \\ &= 1 - \mathbb{P}\left[K < m|E=1\right] \\ &\geq 1 - \exp\left(-2(\mathbb{E}\left[K|E=1\right] - m)^2/\pi R^2 d\right) \end{aligned} \tag{2.13}$$

for all positive $m \leq \mathbb{E}\left[K|E=1\right]$.

The inequality in Eq. (2.13) comes directly from the special case of the Chernoff/Hoeffding bounds for the Poisson Binomial Distribution tail [30]. Because $0 \leq p_i \leq 1$, the denominator inside the exponential is the number of expected sensors in the aggregation window. It is an upper bound, as m is expected to be smaller than $\mathbb{E}\left[K|E=1\right]$.

Probability of false positive. Similarly, we denote the sensor false detection as q, where $q = \mathbb{P}\left[d=1|E=0\right]$. The probability of receiving exactly k sensor detections within the same amount of time from noise is

$$\mathbb{P}\left[K=k|E=0\right] = \sum_{A\in F_k} \prod_{i\in A} q_i t \prod_{j\in A^c} (1-q_j t)$$

The analysis assumes that all sensors have the same background noise q, $q_i = q, \forall i = 1,\ldots,n$. The expected value K is simply the mean of a binomial distribution

$$\mathbb{E}\left[K|E=0\right] = \sum_{i}^{n} q_i = \int_{r=0}^{R} (2\pi r d)\, q\, \mathrm{d}r = \pi R^2 dq$$

Since the system declares detection when receiving $K > m$ sensor detections, the probability of false detection is

$$\begin{aligned} pFP &= \mathbb{P}\left[K \geq m|E=0\right] \\ &\leq \exp\left(-2(\mathbb{E}\left[K|E=0\right] - m)^2/\pi R^2 d\right) \\ &= \exp\left(-2\left(qd\pi R^2 - m\right)^2/\pi R^2 d\right) \end{aligned} \tag{2.14}$$

for all positive $m \geq \mathbb{E}[K|E=0]$. Here, again, Chernoff/Hoeffding bound is used to compute the tail probability. It is a lower bound because m is expected to be larger than $\mathbb{E}[K|E=0]$.

Selecting detection threshold. The upper bound of pFP in Eq. (2.14) is a function of detection threshold m. By rearranging the equation, we can compute the minimum detection threshold to ensure that a maximum tolerable false positive rate is not exceeded. Let this maximum pFP be g, where $0 \leq g \leq 1$. We have

$$pFP \leq \exp\left(-2\left(\pi R^2 dq - m\right)^2 / \pi R^2 d\right) = g$$

rearranging this to pull out m, we get

$$m = \left\{\pi R^2 dq - R\sqrt{\frac{-\ln(g)d\pi}{2}},\ \pi R^2 dq + R\sqrt{\frac{-\ln(g)d\pi}{2}}\right\}$$

Because $m \geq \mathbb{E}[K|E=0] = \pi R^2 dq$, we have

$$m = \pi R^2 dq + R\sqrt{\frac{-\ln(g)d\pi}{2}} \tag{2.15}$$

Plugging Eq. (2.15) into Eq. (2.13), and substituting R with vt we have the upper bound for the probability of true positive.

$$\begin{aligned} pTP &\geq 1 - \exp\left(-2(\mathbb{E}[K|E=1] - m)^2 / \pi R^2 d\right) \\ &= 1 - \exp\left(-2\left(\frac{\pi\alpha d}{\beta}\left(1 - e^{-\beta R^2}\right) - \left(\pi R^2 dq + R\sqrt{\frac{-\ln(g)d\pi}{2}}\right)\right)^2 / \pi R^2 d\right) \\ &= 1 - \exp\left(-2\left(\frac{\pi\alpha d}{\beta}\left(1 - e^{-\beta v^2 t^2}\right) - \left(\pi v^2 t^2 dq + vt\sqrt{\frac{-\ln(g)d\pi}{2}}\right)\right)^2 / \pi v^2 t^2 d\right) \end{aligned} \tag{2.16}$$

□

2.3.4 Optimal Integration Window Size

To get an idea of the bound, the theoretical pTP bound is plotted as a function of event speed v (Fig. 2.8(a)) and as a function of aggregation time t (Fig. 2.8(b)), using a false positive bound $g = 1 \times 10^{-7}$, which is roughly equivalent to one false alarm a month. As expected from Fig. 2.8(a), when the integration window size is kept constant, the event becomes easier to detect when the signal spreads out more rapidly. However, the peak in Fig. 2.8(b) suggests that while keeping the speed constant, there is an optimal integration window size, or in other words, an optimal area size to be aggregated. This result may be counter intuitive, as having more information should always

improve the detection performance. Indeed, this result is an artifact from ignoring the magnitude parameter in the attenuation model.

The probability p_i that a sensor s_i detects an event is determined by the manifestation of the event $\phi(f)$ and noise q_i.

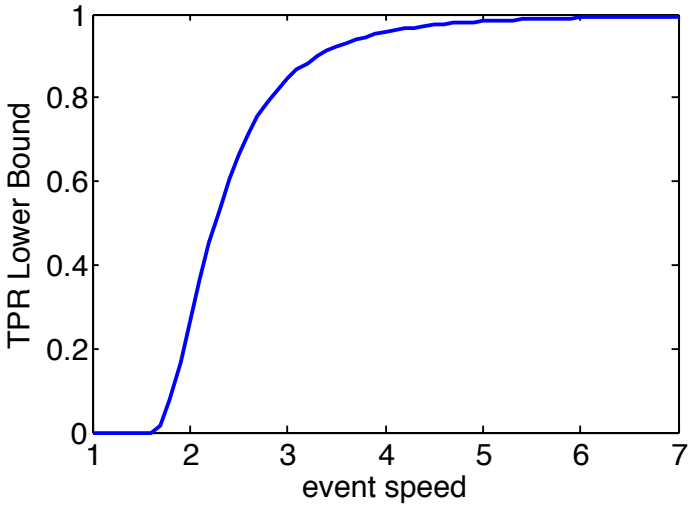
$$\mathbb{P}\left[X = 1 | E = 1\right] = \phi(f) + q_i - \phi(f)q_i$$

$\phi(f)$ depends not only on the distance to the event but also on the magnitude of the event. For a large event, the probability of an event detection is higher for larger window because the impact propagates over a larger region. Likewise, the probability is higher for shorter window when the magnitude is low because of the impact falls off and the difference between the impact and noise cannot be differentiated. An optimal integration window size can be derived by optimizing Eq. (2.13) for a specific sensor layout and magnitude. Since magnitude information cannot be assumed a priorly, a good choice of window size is the one that optimizes Eq. (2.13) for the minimum magnitude one wishes to detect.

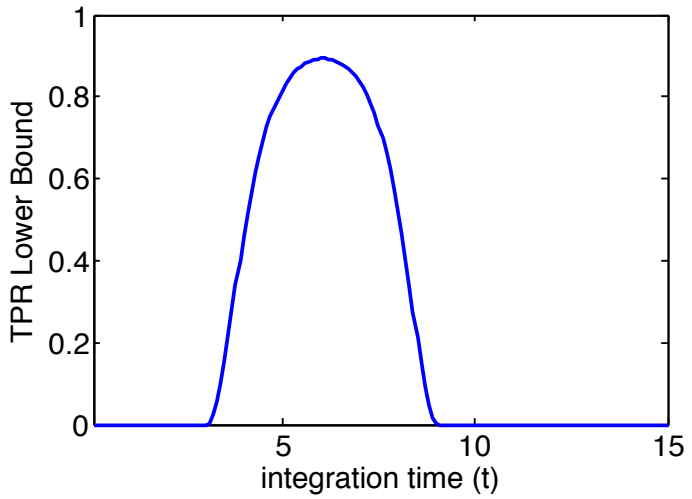
2.4 Discussion

Geospatial events are complex phenomena. Detection performance of geospatial events is affected by many variables. Besides the obvious parameters, such as the number and the quality of sensors, other parameter variables include location of the sensors, sensor and server timing error, event magnitude and location, and speed of the event's manifestation. Depending on the event type, other environmental variables, such as temperature, moisture, wind speed, man-made structures can also play a crucial role. To capture all the complex interplays of variables in an analysis of detection performance is impossible.

To acquire comprehensible results, the analyses included in this chapter made several assumptions on sensor layout, and event location and magnitude. Some of these may be oversimplification; for example the event does not always occur within the boundary of the network with a fixed magnitude and speed. However, without such assumptions the analyses become far too complex to be meaningful. Although theoretical results with these restrictions may be considered purely academic, we argue that in terms of validation of intuitions and understanding performance tradeoff, they provide very practical and valuable insights into the design of an event detection system.



(a) Increasing speed



(b) Increasing integration time

Figure 2.8: Theoretical network detection bound. (a) TPR lower bound as a function of event propagation speed. $R = 50$, $d = 0.006$, $q = 0.01$. (b) TPR lower bound as a function of integration window size. $v = 5.5$, $d = 0.014 \approx 100$ sensors, $q = 0.01$.

Chapter 3

Estimating the Parameters of Geospatial Events

In many event detection problems (e.g. entry detection), the goal of the algorithm is to identify whether an event has taken place. In such cases, a simple low pass filter that compares recent observations to less recent ones may suffice to distinguish between the null hypothesis $\mathcal{H}_0$ and the alternative $\mathcal{H}_1$. But for most warning applications for geospatial events, reliable rapid detection depends on the system's capability to correctly estimate event related parameters (e.g. origin location and magnitude) in order to take proper actions. For example, a region should not receive a warning for an earthquake that will not cause significant shaking in that region. Fast and accurate estimation allows the community more time to respond.

Classical and Bayesian approaches. Most detection and parameter estimation algorithms can be categorized into two approaches — classical and Bayesian. Classical approaches focus on deriving estimates that best fit some parametric models of the events. Such approaches include inverse-law inference [21, 71], Maximum Likelihood Estimator [39], 2-dimensional least squares fitting (LS) [44], sequential probability testing (SPRT) [49, 75, 74, 73], optimized fitting through genetic algorithm [80], and learning inverse relationship through neural network [10, 51]. These methods often produce single point solutions without giving a well-defined measure of its quality. In addition, the estimates are often discontinuous, because estimates from a previous time step do not affect the estimates at a later time. There is also no natural way to incorporate prior knowledge about event parameters.

Bayesian approach, on the other hand, is a unifying approach that naturally incorporates prior information to produce a probabilistic estimate. Some of these methods include Bayesian posterior estimation [16, 87, 64, 25, 24], Bayesian estimation with classification [69], and Extended Kalman Filter and its variants [40]. Rather than a single point solution, Bayesian approach produces a posterior probability distribution, and each estimate is associated with a confidence measure. Incorporation of appropriate prior information also generally leads to better estimates and faster convergence time with fewer observations.

Bayesian prior selection. The accuracy of detection in a given time interval depends on prior distributions of parameters. Prior distribution can be constructed from historical data. For example, the distributions of hypocenters can be constructed based on known fault lines. However, selection of appropriate priors is challenging when the distribution is unknown and/or the space is continuous and unbounded. The former problem can usually be solved with a flat prior, but the latter requires the distribution space to be truncated. An approach to resolve this issue as follows: Begin with a flat prior h over a very large space, where the *a priori* probability of data outside the space is extremely small. The posteriori probability will also be flat because the probability distribution is spread over a large space. We use a maximum *a posterior* probability similar to a maximum likelihood estimator, in the following way.

After data has arrived for some time t, we compute the posterior distribution $f(t)$ and determine the points of maximum *a posterior*. Now construct a new prior $g(t)$, with probability distributed around these points, and falling away, relatively rapidly, for points further away from the maximum *a posterior* regions. The new prior will have a single peak or a few peaks, and will not be flat as h is. Now continue with Bayesian computation, starting with the new prior, as additional data streams in.

This idea is equivalent to the *integrated* approach that we present later, i.e. using a classical maximum likelihood estimator after data arrives for time t, and then using the resulting estimate to construct a prior for a Bayesian calculation that follows after time t. It is similar to the two-level Bayesian inference described by MacKay, i.e. first select the most probable parameters for each hypothesis model, then infer which model is most plausible given the data [61]. However, unlike MacKay's method, which still requires a finite number of candidate parameters to be computable, our simplifying approach is efficient but may lead to inaccurate estimate for several reasons. For example, the maximum likelihood estimator may be far from the true maximum likelihood region, or the posterior distribution may not be smooth. We show empirically that for our class of geospatial problems, this approach works well.

Contributions and chapter organization. This chapter presents a suite of classical (Sec. 3.1) and Bayesian (Sec. 3.2) approaches on parameter estimation with two applications in mind — radiation and earthquake. Some of these approaches have been considered by others in similar formulations, but we further supply them with rigorous theoretical analyses (Sec. 3.2.2). Furthermore, we address the difficult problem of prior selection and introduce an intuitive integrated algorithm that combines the strengths of both classical and Bayesian approaches (Sec. 3.3). The effectiveness of the algorithm is validated through extensive simulations, as well as testbed experiments (Sec. 3.3.1). These results have appeared in the following conference proceedings [59, 56].

3.1 Classical Algorithms

Classical algorithms often involve optimum fitting of observed data to a parametric model. These techniques include back projection, triangulation, linear best fit, and maximum likelihood. This section presents two classical algorithms tailored for the two applications — radiation and earthquake. These algorithms are ideal candidates for the integrated algorithm that will be introduced later in Sec. 3.3.

3.1.1 Radiation — Dynamic Sensor Grouping

Radiation sensors detect photons that travel at the speed of light and it is safe to assume that there is no time difference between the start of the event t_0 and the time when a sensor receives the first measurement. In detecting a radiation source, the key parameters to estimate include background noise level, source location, source intensity, the type of source, and the probability that a source is present. Coarse estimates of these parameters can be obtained following the idea of maximizing signal to noise ratio (SNR) through sensor grouping, as briefly introduced in Sec. 2.2.1. The algorithm (denoted as the `K-sigma` algorithm) is derived from the following observations.

Dynamic sensor grouping. Since the SNR decreases as a function of the distance between a sensor and a source, combining measurements by treating all sensors equally may decrease the overall SNR [58] because some sensors are closer to the source and some sensors are further away. However, if we can group the measurements in terms of distance to the source, then we are guaranteed to maximize SNR. But how should sensors be grouped if the source location is unknown? We solve this problem by exploring all reasonable sensor groupings by proximity using Delaunay triangulation — an efficient technique from computational geometry that partitions a space into triangles, while maximizing the minimum angle of the triangles [28]. Fig. 3.1 is an example of a triangulations done with a network of nine sensors.

Estimating background radiation. With a sensor grouping, we estimate the background noise, assuming that it is uniform within the region. The sensor's flux drops off faster than $1/r^2$ due to absorption of photons in the air, where r is the distance to the radiation source. As a result, sensors far from a source receive negligible flux from a threat. We estimate the background flux throughout the region by computing the average rate of photons received by these distant sensors. This can be done by breaking the region up into quadrilateral cells (pairs of triangular cells that share a common edge), where each cell is identified by the sensors at its four vertices. The total photon count received in each cell is called its *cell count*. We use the cell with the lowest cell count, after correction for order statistics, as the background rate estimate Γ for all sensors. This background noise estimate is used when estimating other parameters.

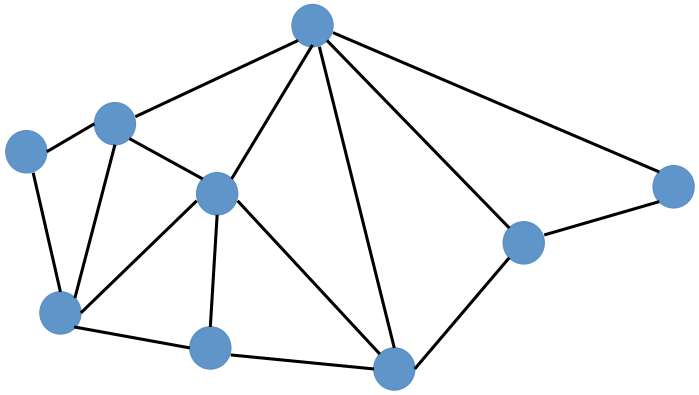


Figure 3.1: Snapshot of Delaunay triangulation as part of the kSigma routine. The partitioning is used to compute the sensor groups for data aggregation. The triangulation can be efficiently computed in real time.

Estimating whether a source is present. The relative source location in the sensor network can be coarsely categorized as follows: the source is close to (a) one of the sensors, (b) an edge between two sensors, (c) the center of a triangle, and (d) the center of a quadrilateral cell of sensors as illustrated in Fig. 3.2. If a source is near the center of a cell, then the average counts measured by the sensors at the four corners of the cell are likely to be higher than the counts from the background. Likewise, if a source is very near a sensor then the photon count measured by that sensor is likely to be higher than the count from the background. So, we estimate the average counts from each *single* sensor, the pair of sensors along each *edge*, and the sensors at the corners of each *triangle* and each *cell*. We compute the number of standard deviations, called *kSigma values*, of the measured counts from the estimated counts if only the background were present; and we compute kSigma for each and every group (singleton, edge, triangle, quad) in the field. Specifically we compute for each group

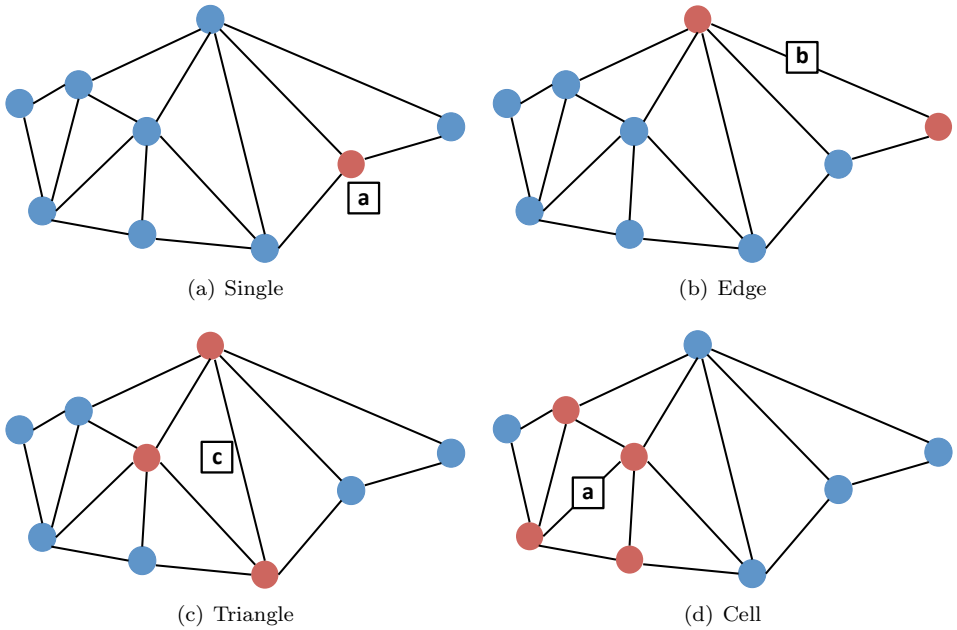
$$kSigma = \frac{N_g - \Gamma}{\sqrt{\Gamma}} \tag{3.1}$$

where N_g is the group's aggregate radiation count in time T, and Γ is the estimated aggregate count in the group from only the background in time T. The aggregate radiation count in a group is a Poisson process whose standard deviation is $\sqrt{\Gamma}$. $kSigma$ is thus the number of standard deviations that the measured group count is from background noise.

The kSigma values are then corrected using order statistics to account for the bias in ordering the groups by aggregate counts. The division factors used for correction are (for $N = 9$ sensors)

$$K_1 = 1.01,\ K_2 = 1.1,\ K_3 = 1.3,\ K_4 = 1.5$$

for singleton (K_1), edge (K_2), triangle (K_3), and quad (K_4). For example, if the aggregate counts for a quad of sensors is 81, and the calculated background at each sensor is 9, then the uncorrected



(a) Single (b) Edge

(c) Triangle (d) Cell

Figure 3.2: Examples of `K-sigma` grouping from the Delaunay triangulation with four possible source positions relative to the sensor network illustrated.

kSigma for the group would be $(81 - 4 \times 9)/\sqrt{9} = 15$ and the corrected value would be $15/K_4 = 10$. To estimate whether a source is present, we apply a threshold τ to the corrected kSigma values for all the groups. The threshold is chosen to maximize the detection rate (or *true positive rate*) while satisfying a constraint on the maximum false positive rate.

Estimating source intensity and location. Let x be a d-dimensional vector representing a point in space. Consider an experiment conducted over an interval of duration T. Let Γ be the expected number of photons due to background, and n_j the number of photons measured at sensor j in this interval. The flux at a point decreases roughly as the square of the distance, ignoring absorption in the air. If there is a source with intensity μ present at location x, then $\lambda_j(x)$, the number of photons from the source at location x measured at sensor j in the interval T, is approximately

$$\lambda = \frac{C\mu T}{|S_j - x|^2}$$

where C is a constant of proportionality that depends on the sensitivity of the sensor.

If $n_j > \Gamma$ then the difference $n_j - \Gamma$ is attributed to a source of intensity $\mu(x)$ that can be at any location x

$$n_j - \Gamma = \lambda_j(x) = \frac{C\mu(x)T}{|S_j - x|^2}$$

For convenience, we define effective source intensity as $\upsilon_j(x)$

$$\upsilon_j(x) = C\mu(x)T = (n_j - \Gamma)|S_j - x|^2$$

To estimate the source position, we make use of the sensor quad that exhibits the largest value of kSigma. We compute $\upsilon_j(x)$ for all sensors j in this quad, for all locations x. The location that is most consistent with the observed sensor readings is most likely to be the source location. To quantify this, we compute a variance estimate

$$M(x) = \sum_{j=1}^{m} (\upsilon_j(x) - \overline{\upsilon(x)})^2$$

where m is the number of sensors and $\overline{\upsilon(x)}$ is the average $\upsilon_j(x)$ for j ranging over the vertices of the quad. We postulate that the likelihood of a source at a point x is inversely proportional to this variance function M. The source location estimate $\hat{x}$ is thus

$$\hat{x} = \arg\max_x \frac{1}{1 + M(x)}$$

Once the source location has been determined, the estimation of its intensity $\hat{\mu}$ is straightforward: it is derived using the distance of each sensor from the source and the elapsed time T.

$$\hat{\mu} = \frac{1}{mCT} \sum_{j=1}^{m} \upsilon_j(\hat{x})$$

3.1.2 Seismology — Arrival Time Based Epicenter Estimation

Earthquake waves, unlike photons, travel at a finite speed of 3-8 km/sec. The delay between the event start time t_0 and the time at which a sensor detects a seismic wave is roughly linear with the distance. This property leads to the design of a simple arrival time based algorithm to estimate the earthquake epicenter.

Comparison to existing methods. Similar algorithms of varying complexity have been studied in the seismology literature, such as the continuous least square method [71, 72] and adaptively damped least square method [54]. However, the approach presented here is targeted for detecting a wide range of geospatial events in dense noisy networks rather than professional networks, and is designed to be fast, simple, scalable and tolerant to noise. To make such generalization possible, the method makes a few simplifying assumptions. One of these is that the model ignores the vertical coordinate of the event initiation location. It also assumes that the event propagation velocity is time- and space-invarying. Both of these are coarse approximations to the actual phenomenon.

A simple and fast algorithm. Let (x_0, y_0, t_0) be the event origin point in the space-time dimension. The time t_j at which the event first reaches a sensor j at location (x_j, y_j) is $t_j = \frac{d_j}{v} + t_0$, where d_j is the distance of sensor j from the event origin, and v is the velocity of the propagation. Therefore:

$$t_j = \frac{\sqrt{(x_j - x_0)^2 + (y_j - y_0)^2}}{v} + t_0$$

If sensor j generates a pick corresponding to observation of the event, let the time at which it generates the pick be $\hat{t_j}$. The error e_j in the time that sensor picks, according to this model is: $e_j = \hat{t_j} - t_j$. Let z be the sum of error squared, $z = \sum_j e_j^2$. Our goal of this analysis is to compute (x, y, t) to minimize z given the set $(x_j, y_j, t_j), j = 1, \ldots, n$ of sensor measurements while discarding outlier picks.

A local minimum of z is the solution to the nonlinear equations:

$$\sum_j \frac{(t_j - \frac{d_j}{v} - t)(x_j - x_0)}{d_j} = 0$$

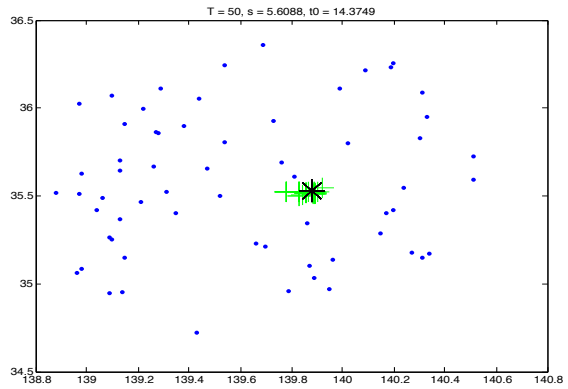
$$\sum_j \frac{(t_j - \frac{d_j}{v} - t_0)(y_j - y_0)}{d_j} = 0$$

where $t = t_j - \frac{d_j}{v}$.

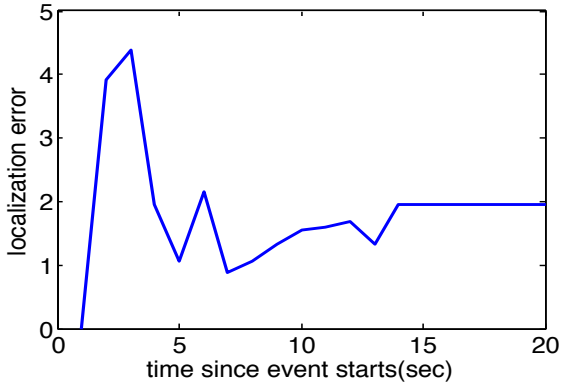
Case Study - Japan seismic network. We analyze two independent events from publicly available seismic data from Japan, one with an inland epicenter (Fig. 3.3(a) and Fig. 3.3(b)), the other with an offshore epicenter (Fig. 3.3(c) and Fig. 3.3(d)). These high quality sensors are roughly 20 km apart. The parameters to be estimated are [lat, lon, speed, start time] = $[x, y, v, t_0]$. As demonstrated by the localization error plots, the simple algorithm performs fairly well in comparison to official epicenter estimates determined by the Japan Meteorological Agency based on much more sophisticated model.

Case Study - Community seismic network. The same algorithm is applied to community sensors. Fig. 3.4(a) plots the 185 sensors from the Community Seismic Network in southern California and four events that were recorded between April and September 2012. Location estimation is carried out immediately after detection with a slightly more sophisticated wave propagation model suitable for southern California. The model produces arrival time for P-wave and S-wave using distance and depth as input. Since the model can be stored as a table, lookup time is constant. The parameters to be estimated are [lat, lon, depth, start time] = $[x, y, z, t_0]$. The localization error with respect to the epicenters determined by USGS are plotted in Fig. 3.4(b) as a function of time since the event was detected.

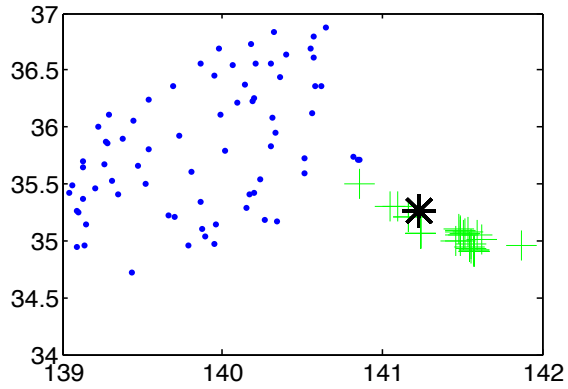
Fig. 3.4(c) and Fig. 3.4(d) show the algorithm fit at 0 and 5 seconds after detection during the



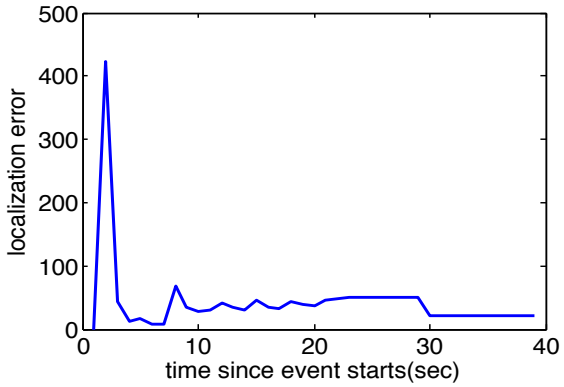
(a) Inland epicenter estimates



(b) Error in epicenter estimate for an inland epicenter



(c) Offshore epicenter estimates



(d) Error in epicenter estimate for an offshore epicenter

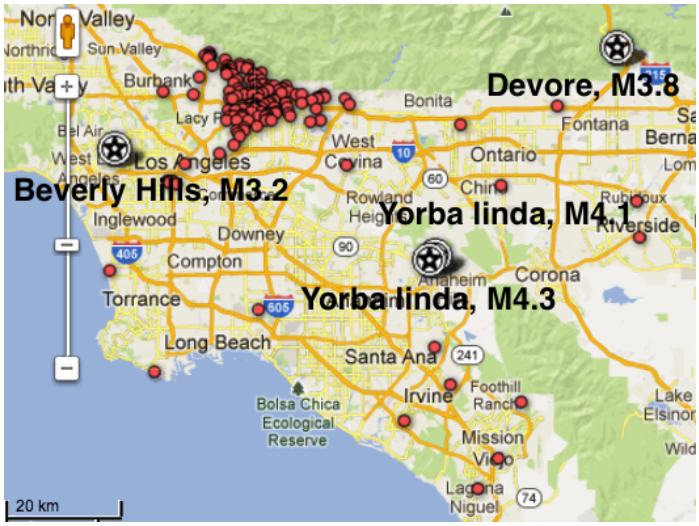
Figure 3.3: Validation of the simple arrival time algorithm with Japan seismic network. Blue dots represent sensor locations. The black cross is the epicenter identified by JMA, whereas green crosses are epicenters determined by the simple algorithm at each time step. The simple approach performs fairly well with this high quality data set.

Yorba Linda2 event. It is clear that, even with the high level of sensor noise (premature picks, continuous picking, and noisy picks), the simple algorithm is able to produce reasonable estimates within a few seconds. It should be noted though that the estimates are expected to be worse for events further away from the network because actual wave propagation is nonlinear with distance along the surface of the globe; an example is the Devore event in Fig. 3.4(b).

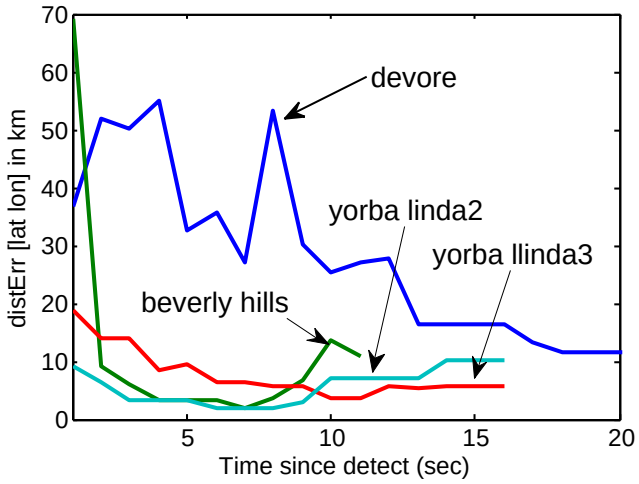
3.2 Bayesian Algorithm

Bayesian approach has advantages over classical methods. For one thing, rather than a single point estimate, it generates a posterior probability distribution that is useful in answering questions regarding how confident we are about the estimate and when the estimate will converges. Furthermore, with appropriate prior distributions, the Bayesian approach converges more quickly.

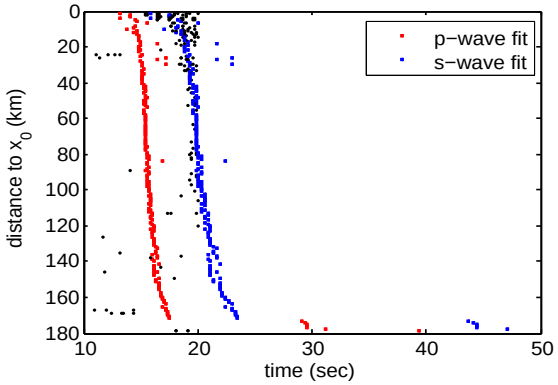
This section presents the Bayesian approach using radiation estimation as an example. It then



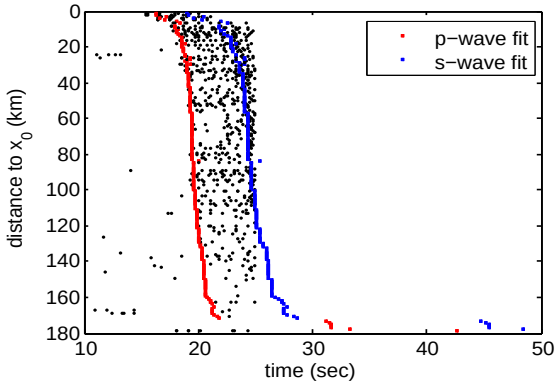
(a) Map of CSN sensors and the location of the four recorded events



(b) Localization error for the four events.



(c) Yorba linda2, 0 second after detection



(d) Yorba linda2, 5 seconds after detection

Figure 3.4: Arrival time based parameter estimation with community sensors. (c) and (d) are fits computed by the simple least square methods. Black dots indicate sensor picks ordered by the sensor distance to the estimated epicenter. Red and blue dots are the expected P-wave and S-wave arrivals. The plots indicate that the simple algorithm performs well even with noisy sensor data.

continues to present a general performance analysis based on the Bayesian approach for all geospatial events in terms of the speed and intensity of the event, and the number of sensors.

3.2.1 Bayesian Estimation Example

Bayesian algorithms compute an *a posteriori* probability distribution based on a given prior distribution and the likelihood values calculated from measurements. Here we elaborate these terms using radiation estimation as an example.

Let θ be the vector of parameters that we want to estimate and $\mathbf{z}$ the network observations. Assuming a prior distribution π_0 for θ, the posterior distribution of θ is:

$$P(\theta|\mathbf{z}) = \frac{P(\mathbf{z}|\theta)\pi_0(\theta)}{P(\mathbf{z})} = \frac{P(\mathbf{z}|\theta)\pi_0(\theta)}{\int P(\mathbf{z}|\theta)\pi_0(\theta)\ \mathrm{d}\theta} \tag{3.2}$$

where $P(\mathbf{z}|\theta) = L(\theta;\mathbf{z})$ is the likelihood of observing $\mathbf{z} = \{z_1, z_2, ..., z_m\}$ photons at sensor $j = 1, 2, ..., m$ in time interval $[0, t]$. Hereafter, we assume that the time interval t during which measurements are made is the same for all sensors, and therefore omit t writing z_j rather than $z_j(t)$. L is found by the sensor measurement equation as:

$$L(\theta;\mathbf{z}) = \prod_{j=1}^{m} f(z_j;\Lambda_j(\theta)) \tag{3.3}$$

In the case of radiation detection, sensors measure the number of photons striking the sensors. Photon emission is a Poisson process, therefore

$$f(z_j;\Lambda_j(\theta)) = \frac{\Lambda^{z_j} e^{-\Lambda_j}}{n!}$$

is the Poisson probability mass function where $\Lambda(\theta)$ is the expected number of photons measured at sensor j in time t. θ includes parameters such as source strength μ, source location D, and expected number of photons Γ from the background in time t. The likelihood function can be modified to incorporate additional information; for example photon energy [15].

In the scenario where the background noise is profiled beforehand, given the *a priori* probability $\pi_0(\theta)$ and the measured data n, the algorithm computes the *a posteriori* probability that a source of intensity υ is located at position x. The posterior probability that a source is present is the summed probability over all x and υ. If the algorithm must make a binary decision — a threat is present or is not present — then it decides that a threat is present if the posterior probability that a source is present exceeds a threshold where the threshold is determined by the tolerance for false positives.

3.2.2 General Parameter Estimation Analysis

Radiation is a special case of geospatial event where the event signal reaches all the sensors almost instantaneously. In this case, raw measurements are essential for parameter estimation. For most geospatial events (e.g. earthquake, flood, air pollution) the signal travels at a finite speed and the detection time difference between each sensor can be used to estimate the event parameters. The heuristic described in Sec. 3.1.2 is one such example. With this property, the sensor-server communication can be reduced to one-bit information only when the sensor detects something abnormal. This is extremely powerful for large scale community sensor networks when it is infeasible to stream raw measurements from all the sensor at all times. More discussion on large scale network is included in Chapter 6.

The same property can be applied to Bayesian estimation, that is, to compute the probability of each possible estimate given a sequence of sensor detections and their timestamps. In this case, the posterior probability distribution is affected by the speed of the event, the number of sensors, and the timing errors within each sensor detection. In the following section we analyze and bound the performance of Bayesian parameter estimation for general geospatial events.

3.2.2.1 Preliminaries

The following analysis is based on these assumptions

- The event can be modeled as a point source with origin location x_0 and start time t_0.
- The event propagates through the network at a constant speed v.
- The sensor timing error can be modeled as a 0-centered Gaussian random variable $\mathcal{N}(0;\sigma)$ (note that, in reality, this is rarely the case for individual sensors).
- The prior probability distribution is uniform, i.e. the event can start anywhere in space-time.
- The n sensors in the network are semi-randomly distributed (e.g. they do not all lie on a single point or single line).
- The magnitude of the event is large enough such that all sensors detect it with probability 1.0.

3.2.2.2 Estimate Expected Error

Given that an event has been detected, the posterior probability of event parameters can be computed with a list of sensor detection data around the detection time. The Bayes posterior distribution of the event origin (x) and origin time (t) is:

$$\mathbb{P}\left[x,t|\cdot\right] \propto \prod_{i=1}^{n} \left[\mathbf{1}_i p_i + (1-\mathbf{1}_i)(1-p_i)\right]$$

$\mathbf{1}_i$ is the indicator function. $\mathbf{1}_i = 1$ when sensor i at location s_i has at least one detection for the event, propagating at speed v, in the time interval

$$\left[t + \frac{\|s_i - x\|}{v} - k,\ t + \frac{\|s_i - x\|}{v} + k\right]$$

k is the window width. p_i is the probability of detection for a sensor at distance r_i from the event H. $p_i = \phi(H)$.

Assuming that the magnitude of the event is large enough, such that for all sensors, $p_i = 1, \forall i = 1, \ldots, n$, and the detection threshold is high enough that there are no detections due to noise, we have the following results for the variance of location and time estimates.

Theorem 2. *Under the aforementioned assumptions, the expected error of the Bayesian posterior estimates on event location $x_{err} = |x - x_0|$ and in event origin time estimate $t_{err} = |t - t_0|$ is bounded below in terms of sensor timing error (σ), number of sensors (n), and the speed the event travel at (v).*

$$\mathbb{E}\left[x_{err}\right] \geq \frac{\sigma v}{4}\sqrt{\frac{\pi}{2n}} \tag{3.4}$$

$$\mathbb{E}\left[t_{err}\right] \geq \frac{\sigma}{4}\sqrt{\frac{\pi}{2n}} \tag{3.5}$$

These results show that the quality of location estimate is influenced by timing error ($\mathbb{E}\left[x_{err}\right] \propto \sigma$) and how fast the event travels ($\mathbb{E}\left[x_{err}\right] \propto v$). The same goes with the quality of time estimate ($\mathbb{E}\left[t_{err}\right] \propto \sigma$). Next we will prove these results by first statin the following lemma with its proof included in the appendix.

Lemma 1. *Let $f(x)$ be the probability density function of a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$. Let $G(x)$ be the area under curve of $f(x)$ in the interval $[x - \sigma,\ x + \sigma]$ then*

$$G(x) = \int_{x-\sigma}^{x+\sigma} f(x)\,\mathrm{d}x = \int_{x-\sigma}^{x+\sigma} \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$$

$$C\, e^{-(x-\mu)^2/2\sigma^2} \leq G(x) \leq C\, e^{-(x-\mu)^2/4\sigma^2}$$

where $C = erf\left(\frac{1}{\sqrt{2}}\right)$.

Proof of Theorem 2. If $\hat{p}$ is the actual time when a sensor first detects the event and p is the timestamp it reports, then the timing error $q = p - \hat{p} \sim \mathcal{N}(0; \sigma)$, assuming the error can be modeled as Gaussian. Let $k(\cdot)$ be the time difference between $\hat{p}_{x_0,t_0}$ (expected sensor detection time for an

event at (x_0, t_0)) and $\hat{p}_{x,t}$ (expected sensor detection time for an event at (x, t))

$$k(x,t) = \hat{p}_{x,t} - \hat{p}_{x_0,t_0} = \frac{\|s - x\| - \|s - x_0\|}{v} + (t - t_0)$$

where s is the sensor location, and

$$0 \leq |k| \leq \frac{\|x - x_0\|}{v} + |t - t_0| = \frac{x_{err}}{v} + t_{err} \tag{3.6}$$

The upper bound on k comes from triangle inequality. The probability that p is the detection time for an event (x, t) is the probability that q falls between the interval $[k - m\sigma, k + m\sigma]$ which is a window of arbitrary width $2m\sigma$ centered at $k(\cdot)$. For simplicity, assume $m = 1$ and call this probability $G(k)$. Since the likelihood $\mathbb{P}[\cdot|x,t]$ is the probability that all sensor detections fall within a 2σ window centered around its expected detection time for an event initiated at (x, t), assuming a independent sensor, $\mathbb{P}[\cdot|x,t]$ is the product of $G(k_i), \forall i = 1, \ldots, n$. Assuming uniform prior distribution, the posterior probability $\mathbb{P}[x,t|\cdot]$ is then

$$\mathbb{P}[x,t|\cdot] = \frac{\mathbb{P}[\cdot|x,t]\,\mathbb{P}_0[x,t]}{\int \mathbb{P}[\cdot|x,t]\,\mathbb{P}_0[x,t]\,\mathrm{d}(x,t)} = \frac{1}{S}\,\mathbb{P}[\cdot|x,t] = \frac{1}{S}\prod_{i=1}^{n} G(k_i)$$

Without loss of generality, let $x_0 = (0,0,0)$, $t_0 = 0$, $m = 1$. Assuming the sensors are semi-randomly placed, and observe that $G(k)$ is monotonically decreasing for $k \geq 0$, from Eq. (3.6) $G(k_i)$ can be bounded below by $G(k)$, where $k = \left(\frac{|x|}{v} + |t|\right) \geq 0$.

$$\mathbb{P}[x,t|\cdot] = \frac{1}{S}\prod_{i=1}^{n} G(k_i) \geq \frac{1}{S}\,[G(k)]^n \geq \frac{1}{S}\,C^n \exp\left(-\frac{nk^2}{2\sigma^2}\right) \tag{3.7}$$

The second inequality comes directly from the lower bound in Lemma 1, where $C = erf\left(\frac{1}{\sqrt{2}}\right)$. To simplify the notation, denote $x = |x|$ and $t = |t|$. S is the normalizing factor and can be computed by integrating over all $x \geq 0$ and $t \geq 0$. Using the upper bound in Lemma 1 and substituting k,

$$\begin{aligned}
S &= \int_0^\infty \int_0^\infty [G(k)]^n \;\mathrm{d}x\,\mathrm{d}t \\
&\leq \int_0^\infty \int_0^\infty C^n\, e^{-\frac{nk^2}{4\sigma^2}}\;\mathrm{d}x\,\mathrm{d}t \\
&= C^n \int_0^\infty \int_0^\infty \exp\left(-\frac{n}{4\sigma^2}\left(\frac{x}{v} + t\right)^2\right)\;\mathrm{d}x\,\mathrm{d}t \\
&= C^n \int_0^\infty \sigma v \sqrt{\frac{\pi}{n}}\left(1 - erf\left(\frac{\sqrt{n}t}{2\sigma}\right)\right)\mathrm{d}t \\
&= C^n \frac{2\sigma^2 v}{n}
\end{aligned}$$

Substituting S and k into Eq. (3.7), we get

$$\mathbb{P}\left[x,t|\cdot\right] \geq \frac{n}{2\sigma^2 v}\exp\left(-\frac{n}{2\sigma^2}\left(\frac{x}{v}+t\right)^2\right)$$

Given the posterior probability $\mathbb{P}\left[x,t|\cdot\right]$ we can computed the expected error in location estimate.

$$\begin{aligned}
\mathbb{E}\left[x\right] &= \int_0^\infty x\,\mathbb{P}\left[x|\cdot\right]\,\mathrm{d}x \\
&= \int_0^\infty x\left(\int_0^\infty \mathbb{P}\left[x,t|\cdot\right]\,\mathrm{d}t\right)\,\mathrm{d}x \\
&\geq \frac{n}{2\sigma^2 v}\int_0^\infty x\left[\int_0^\infty \exp\left(-\frac{n}{2\sigma^2}\left(\frac{x}{v}+t\right)^2\right)\,\mathrm{d}t\right]\,\mathrm{d}x \\
&= \frac{n}{2\sigma^2 v}\int_0^\infty x\left(\sqrt{\frac{\pi}{2n}}\sigma\left(1-erf\left[\sqrt{\frac{n}{2}}\frac{x}{\sigma v}\right]\right)\right)\,\mathrm{d}x \\
&= \frac{n}{2\sigma^2 v}\sqrt{\frac{\pi}{2n}}\sigma\frac{\sigma^2 v^2}{2n} \\
&= \frac{\sigma v}{4}\sqrt{\frac{\pi}{2n}}
\end{aligned}$$

Similarly, the expected error in time estimate can be computed

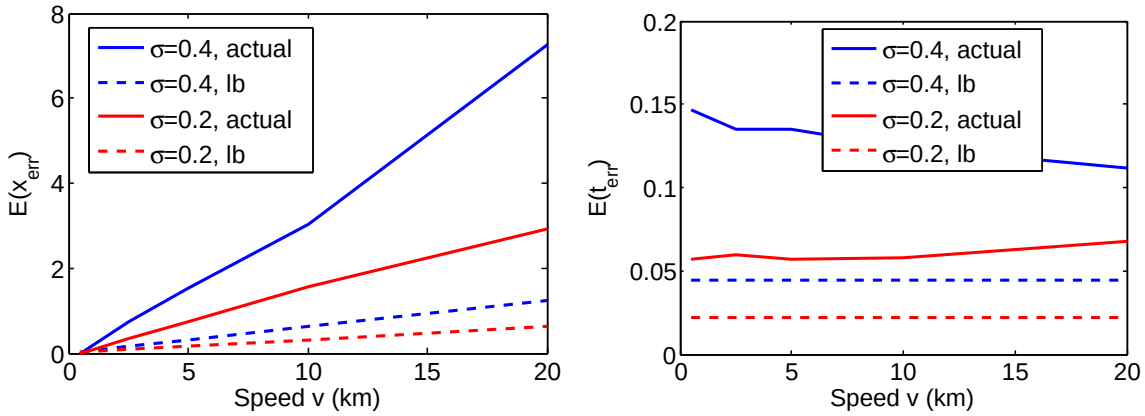
$$\begin{aligned}
\mathbb{E}\left[t\right] &= \int_0^\infty t\,\mathbb{P}\left[t|\cdot\right]\,\mathrm{d}t \\
&= \int_0^\infty t\left(\int_0^\infty \mathbb{P}\left[x,t|\cdot\right]\,\mathrm{d}x\right)\,\mathrm{d}t \\
&\geq \frac{n}{2\sigma^2 v}\int_0^\infty t\left[\int_0^\infty \exp\left(-\frac{n}{2\sigma^2}\left(\frac{x}{v}+t\right)^2\right)\,\mathrm{d}x\right]\,\mathrm{d}t \\
&= \frac{n}{2\sigma^2 v}\int_0^\infty t\left(\sqrt{\frac{\pi}{2n}}\sigma\left(1-erf\left[\sqrt{\frac{n}{2}}\frac{t}{\sigma}\right]\right)\right)\,\mathrm{d}t \\
&= \frac{n}{2\sigma^2 v}\sqrt{\frac{\pi}{2n}}\sigma v\frac{\sigma^2}{2n} \\
&= \frac{\sigma}{4}\sqrt{\frac{\pi}{2n}}
\end{aligned}$$

□

3.2.2.3 Simulation Validation

To quantify the impact of the event and network parameters on the Bayesian estimates, and to evaluate the quality the bounds in Theorem 2, we run simulations with varying event propagation speed v and varying sensor number n.

Varying event speed. We carry out simulations with $n = 16$ sensor uniformly distributed in a 100 x 100 km^2 region. An event can initiate at any location at anytime. In each simulation run, the error in Bayesian posterior estimate for event location $|x - x_0|$ and event starting time $|t - t_0|$ are



(a) (b)

Figure 3.5: Simulation results showing (a) $\mathbb{E}[x_{err}]$ and (b) $\mathbb{E}[t_{err}]$ as a function of event speed v, $v = [0.5, 2.5, 5, 10, 20]$. The results are compiled from 1,000 runs with 16 sensors placed uniformly in a 100 x 100 km^2 area. An event can initiate at any place and time. The lower bounds predicted by Theorem 2 are plotted in dotted lines for reference.

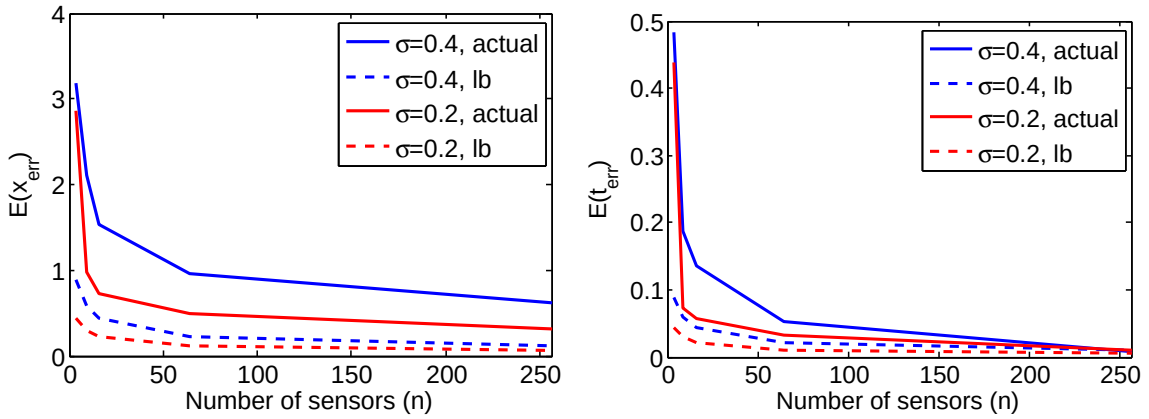
computed. The same set of experiments are repeated for event speed $v = [0.5, 2.5, 5, 10, 20]$ with two values of timing error, $\sigma = 0.2$ and $\sigma = 0.4$. The average errors are shown in Fig. 3.5 as a function of speed v. The lower bounds as derived in Theorem 2 are plotted as dotted line for reference. It is to no surprise that the expected error in location estimate increases as v increase. It is also clear that the bounds capture the relationship between parameters.

Varying network size. To evaluate how the network size affects the parameter estimation performance, we repeat the experiments above, but with varying number of sensors, $n = [4, 9, 16, 64, 256]$. Again, each set of experiments is repeated with two values of timeing error, $\sigma = 0.2$ and $\sigma = 0.4$. The average errors are shown in Fig. 3.6 as a function of network size n.

The results on expected estimate errors in Theorem 2 provide the lower bounds on the quality of the Bayesian posterior estimate on event initiation location and time. The simulations suggest that although the bounds may not be tight, they succeed in providing insights into how the event and network specific variables influence the posterior distribution. In fact, it would be difficult to acquire tighter bounds without stronger assumptions on the network layout.

3.3 Integrated Algorithm

The Bayesian approach suffers in computational efficiency when the prior distribution is too flat and broad. This is common when there is little or no prior information about the parameters. Empirically, the problem can be remedied by truncating the distribution, although an undesirable side effect is slow convergence rate and/or poor estimates when the true values fall outside the prior distribution. Can one do better than random guessing in this situation? We propose estimating the



(a) (b)

Figure 3.6: Simulation results showing $\mathbb{E}[x_{err}]$ and $\mathbb{E}[t_{err}]$ as a function of network size n, $n = [4, 9, 16, 64, 256]$. The results are compiled from 1,000 runs with the sensors placed uniformly in a 100 x 100 km^2 area. An event can initiate at any place and time. The lower bounds predicted by Theorem 2 are plotted in dotted lines for reference.

prior distribution with heuristics. This leads to an integrated approach that combines classical with Bayesian algorithms.

Computation of the integrated algorithm is carried out in two steps — coarse estimation using heurisitics and Bayesian update. The same technique can be applied to all estimation problems. In the rest of the section we focus on the example of radiation detection.

We first use the `K-sigma` algorithm as described in Sec. 3.1.1 to estimate the background rate Γ, the probability that an event source is present at a location x, for each x, and the event intensity μ. These estimates are then used to construct the *a priori* distribution for Bayesian algorithm. Results of `K-sigma` estimation reduce the range of values that need to be considered by the Bayesian algorithm, and thus makes the Bayesian calculations tractable without truncating the priors. We give results from several experiments evaluating the integrated heuristic in Sec. 3.3.1 .

3.3.1 Experiments

We test the three approaches to parameter estimation, i.e. classical, Bayesian, and integrated with both simulated data as well as real measurements collected in a laboratory setting. The algorithm performance is evaluated with the following two metrics.

1. The ROC (Receiver Operating Characteristic) curve that shows the relationship between false positive rates (FPR) and true positive rates (TPR),

2. the DOCA (Distance Of the Closest Approach) curve that plots the probability of localizing within a certain error measured by the distance between the true threat location and the estimated location, and

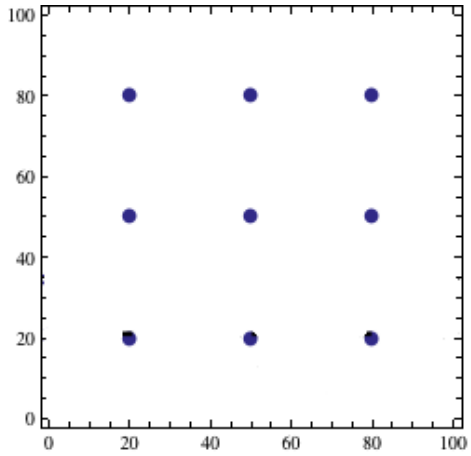


Figure 3.7: Simulation experiment setup. The circles mark the positions of the sensors. Simulated sources are uniformly placed in this field at random.

3. the absolute difference between the estimated magnitude and true magnitude of the threat, if one is present.

3.3.1.1 Simulation

We adopt simulation parameters based on measurements taken by a specific type of CdZnTe radiation sensor.

Setup. We simulate a background of 48 counts per second (cps) and a source of 1200 cps at one meter away. These values are based on measurements made with a real radiation sensors detecting a 1mCi Cesium-137 source. The experiments are set in a 100 x 100 m^2 field with nine sensors placed in a grid formation, as shown in Fig. 3.7. 20,000 simulation runs were carried out for each ROC curve, of which 10,000 were with a source randomly placed in the field and the other 10,000 without. The data is processed at a one-second interval with the following four algorithms:

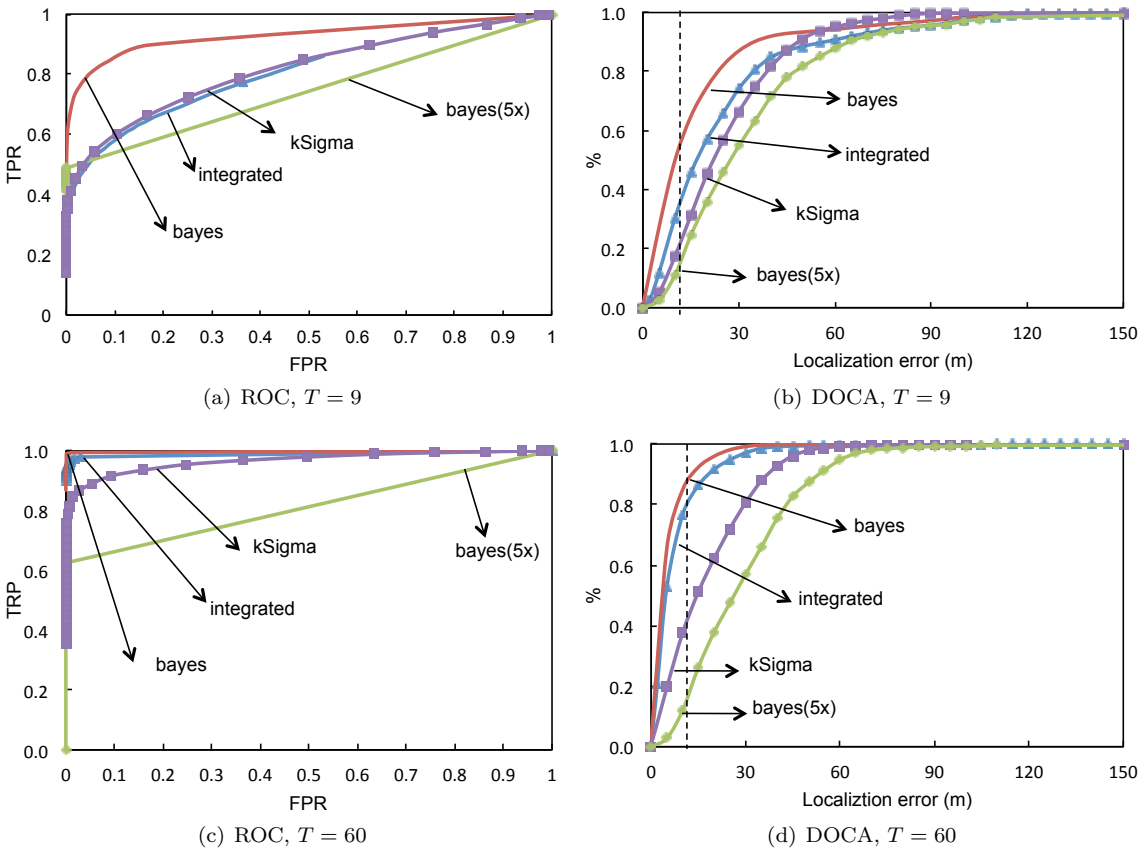
1. Bayesian algorithm with appropriate priors that is centered around the true parameter value (bayes). This serves as the performance upper bound.

2. Integrated algorithm with the prior supplied by `K-sigma` algorithm (`integrated`)

3. Classical algorithm kSigma (`K-sigma`)

4. Bayesian algorithm with inappropriate priors that assume a source five times stronger than is actually simulated (bayes 5x).

Detection probability and location estimates are recorded at T=9 and T=60 seconds. For Bayesian prior, we assume a prior probability that a source is present of 0.1.



(a) ROC, $T = 9$ (b) DOCA, $T = 9$

(c) ROC, $T = 60$ (d) DOCA, $T = 60$

Figure 3.8: ROC and DOCA curves measuring detection and localization performance at (a) T=9 second and (b) T=60 seconds that are compiled from 20,000 simulation runs. The algorithm outperforms both `K-sigma` and Bayesian algorithm with a poorly chosen prior (bayes(5x)). The `integrated` performance approaches Bayesian algorithm with perfect prior (bayes) at T=60 seconds.

Results. We evaluate the algorithm performance using ROC and DOCA curves. The results are plotted in Fig. 3.8. We use the Bayesian algorithm with good priors as the best achievable performance upper bound. At T=9, the `integrated` algorithm has similar detection performance as `K-sigma`; however, there is considerable improvement in localization performance, where the integrated algorithm localizes to within 10m ∼40% of the time as opposed to 20% with `K-sigma` only. At T=60, the gap in performance grows larger. The `integrated` algorithm performs nearly as well as the upper bound both in detection and localization. In all the experiments, the Bayesian algorithm with a inappropriate prior (bayes(5x)) performs much worse than all other algorithms.

These results show that the Bayesian algorithm generally takes much longer to overcome inappropriate priors, and in some cases where the prior distribution does not contain the real value, it never converges at the correct value. In the absence of any information on an otherwise unbounded prior, the integrated approach that combines the classical algorithm with a Bayesian one can outperform a poorly chosen prior.

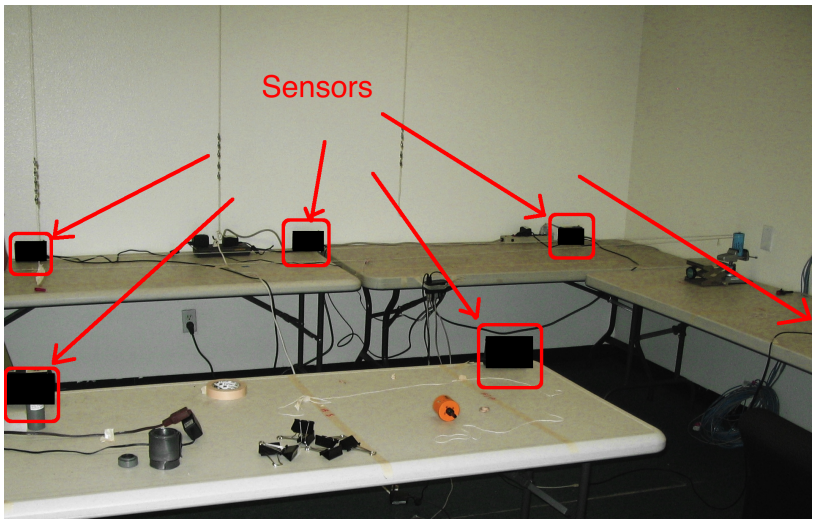


Figure 3.9: Testbed setup in the laboratory. Six radiation sensors are placed in two arrays of three, each circled in red.

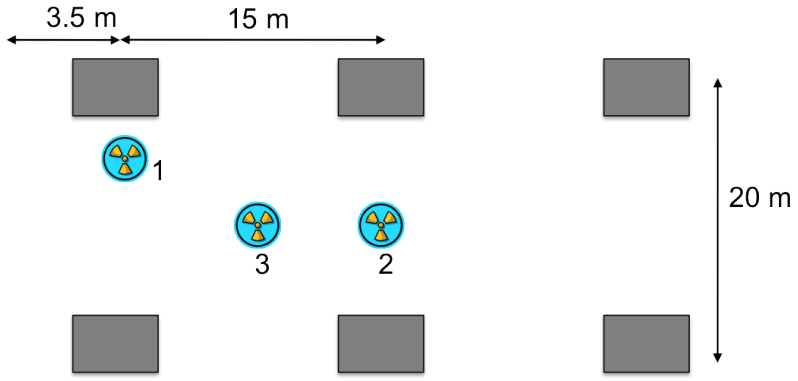


Figure 3.10: Laboratory testbed schematic, showing the six radiation sensors (the grey boxes) and the three source positions.

3.3.1.2 Testbed

Due to space, equipment, and safety limitation, we choose to conduct the experiments in a down-scaled testbed environment.

Setup. We set up a laboratory testbed that consists of 6 radiation sensors arranged in a 3 x 2 grid, as shown in Fig. 3.9. An exempt 9.5 μCi Cesium-137 source can be placed anywhere within or nearby the grid. The spacings between sensors are chosen so that the setup will emulate the detection of a 1 mCi source in a 30 x 20 m field. The schematic is shown in Fig. 3.10. This particular setup is chosen based on on the number of sensors available, the type and strengths of sources, and the amount of time we are able to access the equipment and space. With these constraints, only a small number of representative experiments can be performed.

Experimental data are obtained for the three source positions in Fig. 3.10. Position 1 has the

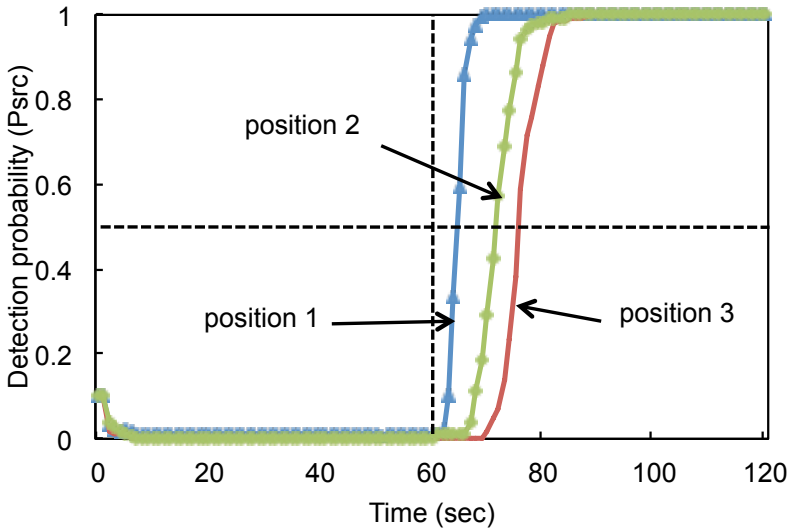
source placed at meter away from a sensor, at (x,y) coordinates (0,19). Position 2 is midway between two sensors at (15,10). Position 3 is in the middle of four sensors at (7.5,10). For each position, we run 100 experiments of 120 seconds each. From T=0-60 seconds, the source is completely shielded and not detectable by the sensors. At T=60 seconds, the shielding is removed and the system observes till T=120 seconds. The data is processed with the Bayesian algorithm with good priors on the source and background strength. The prior probability that a source is present is set at 0.1, and a detection threshold at 0.5. Localization is computed after a detection is made.

Testbed Results. Limited by the number of experiments that can be repeated within a reasonable amount of time, the results are evaluated in terms of averaged detection confidence and localization error over the time span, as shown in Fig. 3.11, instead of ROC and DOCA as used in Sec. 3.3.1.1. The averaged time to detect varies for the three source positions, therefore the localization sequences start at different times. In the first 60 seconds for all 100 runs, no false positives is generated. After the source is unshielded at T=60, it is detected the fastest at Position 1 ($\sim$ 5 sec), followed by Position 2 (10 sec), and 3 (15 sec). This order is inverted in localization: the source is best localized when it is at the center of sensors. This result is not surprising, as measurements from multiple sensors improve target localization. In these experiments, the localization errors converge at about 1 meter. Also observe the dip in localization error for Position 1 at $T = 65$. This is the result from high variance in error among the 100 runs, when localization is attempted with too few data from a single sensor. These results serve to validate the effectiveness of the integrated approach on detecting an static event with networked sensors.

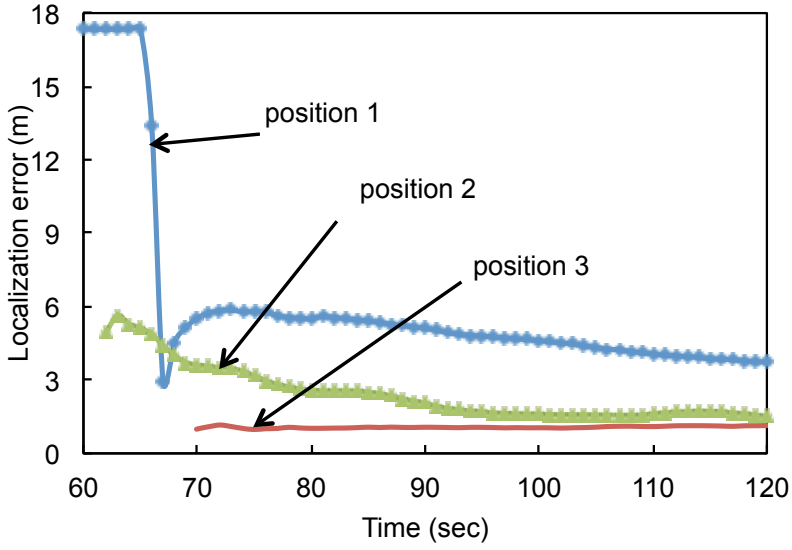
3.4 Discussion

This chapter presents three approaches to parameter estimation for geospatial events, i.e. classical, Bayesian, and integrated, using two event types — radiation and earthquake — as examples. Because the former is a special case of geospatial event (event signal travels at the speed of light and reaches all sensors simultaneously), raw measurements are required for locating the events (streaming mode). General geospatial events, however, can be coarsely located with only the sensor detection time information (compressed mode). We present both a classical optimization-based algorithm and a Bayesian algorithm for the compressed model and prove a variance bound for the Bayesian estimate.

Failure point of the integrated algorithm. Bayesian approach has several advantages over classical methods (heuristics), including producing smooth estimates that can be easily interpreted as probability distribution, and utilizing prior information on the parameters. Unfortunately, its performance suffers severely when the prior distribution is incorrect, e.g. too flat and broad or fails to cover the true values. The integrated approach provides a solution by combining classical and



(a) Detection results



(b) Localization results

Figure 3.11: Laboratory testbed results processed with the Bayesian algorithm with good priors. The results are averaged over 100 test runs.

Bayesian method. Extensive experimentation both with simulated and real measurements were used to validate this approach. While the idea is intuitive and empirically successful, because the priors constructed from classical methods, further investigation is required to determine when it fails, i.e. classical method produces priors far away from the true value.

Analysis on estimate with noise. To simplify the calculation, the theoretical analysis on the quality of Bayesian estimates in Sec. 3.2.2 does not include environmental noise. In reality, sensors are noisy regardless of how high the detection threshold is set. The same analysis in Sec. 3.2.2 can be repeated with a noise term to approximate the effect of environmental noise.

Parameter estimation in decentralized scheme. The algorithms and scenarios discussed in this chapter assume centralized computation. Even the decentralized sensor computation briefly discussed in Sec. 3.2.2 assumes the one-bit sensor detection message will be aggregated at the fusion center. Under some network constraints, such as limited bandwidth and/or limited range, this assumption may be violated. In this case, localized computation is necessary. Chapter 6 will discuss how localized data fusion can be used to compute detection decision and parameter estimation in a large network.

Chapter 4

Mobile Sensors and Mobile Event Sources

Chapter 3 presents three approaches (classical, Bayesian, integrated) to parameter estimation in a traditional setting of sensor networks. In such a setting, both the sensors and events are considered stationary, although the event's manifestation propagates across space. The stationary assumption is a good approximation in some situations, but not for others. For example, in a network that includes cell phone as sensors, a non-negligible number of sensors may be moving during the onset of an event. The event itself may be moving as well. The center of a hurricane can travel at up to 70 mph. Epicenters of large earthquakes are known to travel along a fault line. It is surprise that performance of parameter estimation is affected by these dynamics.

Object tracking. *Object tracking* refers to estimating the location and other parameters of moving objects. When the object's motion is linear and sensor noise is Gaussian, which are the assumptions made with missile tracking, the object location can be estimated using a Kalman filter. Tracking objects with unpredictable movements is more complicated, and often admits no closed form solution. In the field of computer vision, object tracking involves analyzing frame-by-frame images from video, often using a combination of scene parsing and inference through a Hidden Markov Model [89], and fusing data from one or more cameras. In the sensor network literature, object localization has been done through a sequence of triangulations similar to the `K-sigma` algorithm in Sec. 3.1.1 [83, 18]; by combining binary sensor information that indicates whether the object is moving away or towards each sensor [6]; and by organizing sensor nodes as trees to reduce communication cost while triangulating [55], just to name a few. All of these approaches assume that the proximity of the object to a sensor can be accurately approximated, or that the source has a very simple motion (straight line on a street) [48]. In detecting an unknown geospatial event, however, because the sensor can only measure the non-linear, time-varying event manifestation, the proximity of the event to the sensor cannot be determined without knowing the intensity of the event, which is one of the parameters that we are trying to estimate.

Active detection. Active detection is made possible with mobile sensors, e.g. hand-held sensors and sensor-equipped vehicles. In searching for an event source (e.g. radiation, chemical spill) in an open field, real-time sensor redeployment can be done to maximize information gain (minimizing entropy) [78]. In more complex terrains with many obstacles, such as in a city, active detection is extremely challenging due to occlusion and space- and time-varying noise. Cheng *et al.* looked at clustering local sensor data for detection with a network of taxi cabs equipped with radiation sensors [20]. Hochbaum *et al.* employed a weighted concentrated alert approach in the same scenario in order to optimize the detection rate while minimizing the false positive rate [41]. However, these works were based on computer simulations, and not on measurements in the field. They also do not provide adequate insight into tradeoffs between number of sensors, speed of sensor movement, and detection time.

Sensor placement. In some situations, designers face the problem of optimally placing sensors and selecting subsets of sensors for querying. For example, practitioners may want to place sensors to monitor the radiation activity in a critical region. Under budget constraints (i.e. number of sensors), the task can be formulated as an optimization problem. Except for a few special cases, the problem is non-convex and the exact optimization is NP-complete. The earliest work on this topic is the art-gallery problem that assumes each sensor has a fixed disc-like range. The strong assumption transforms the placement problem into a coverage problem that can be approximated with a carefully bounded divide-and-conquer strategy [42]. Other approaches try to relax the assumptions and solve the combinatorial optimization problem with a stochastic search [91, 14]. However, it is difficult to evaluate the quality of the results from these approximate approaches. A first rigorous result on this subject was presented by Krause *et al.* in the work that proves a greedy approach to within (1-1/e) of the optimum by exploiting the submodularity of mutual information [52].

If the goal is to place sensors to maximize *detection speed* and *detection accuracy* for a geospatial event, then the problem has not been addressed by prior work for two reasons: (a) the sensors measure a continuous manifestation of the event and the fixed radius range assumption does not apply here, and (b) the metrics we want to optimize are the detection speed and accuracy that do not translate directly to entropy or mutual information formulation.

Contributions and chapter organization. In this chapter, we explore event and sensor dynamics, and analyze how the network performance changes with traveling sources of events (Sec. 4.1), with mobile sensors (Sec. 4.2), and with customizable sensor placement configurations (Sec. 4.3). As mentioned above, some of these aspects have been studied before, but in settings that cannot be applied directly to detecting geospatial events in the field. This chapter presents rigorous problem and algorithm formulations for geospatial event parameter estimation that are validated by testbed experiments and/or extensive measurement-based simulations.

4.1 Traveling Sources of Events

Geospatial events such as hurricane, earthquake, and radiation contamination are commonly modeled as originated from point, static sources, although sources of these events can travel. Unfortunately, most parameter estimation algorithms in literature do not address such dynamics. This section describes how classical, Bayesian, and integrated algorithm can be modified to track a traveling event.

4.1.1 Modified Algorithms

The three parameter estimation algorithms described in Chapter 3 — classical, Bayesian, integrated — can be modified to address sensor or source mobility. Though in this study, the sensors are assumed stationary for the ease of comparison.

Classical Algorithm. When the event source and sensors are stationary, measurements in the distant past and the more recent past are equally useful; however, when the source or sensors are allowed to move, then measurements in the distant past are less relevant. The modified `K-sigma` algorithm gives higher weight to more recent measurements by aging the relative weights of measurements exponentially with time. At a given elapsed time T, we compute an adjusted weighted sum W of all measurements, as follows:

$$W = \sum_{t=0}^{T} n(t) e^{\frac{t-T}{T_0}} \tag{4.1}$$

where T_0 is a decay constant. If the rate of exponential decay is too large, then results suffer because the usable level of measured data is too low. If the rate of decay is too low, then the results suffer because the measurements do not reflect the prevailing sensor and source positions. The ideal value of T_0 is adjusted based on the estimated source and sensor speeds: the smaller the value, the more sensitive the algorithm is to faster motion.

Bayesian Algorithm. Target motion can be modeled as a state transition prior $P(\theta_t|\theta_{t-1})$ that is multiplied into the posterior, and used as the prior for the next Bayesian update step,

$$P(\theta_t|\mathbf{z}_{t-1}) = P(\theta_t|\theta_{t-1})P(\theta_{t-1}|\mathbf{z}_{t-1})$$

where $\mathbf{z}$ is the sensor observations.

This is a standard procedure in *Kalman filter* and *Particle filter*. In the application of robot localization, missile tracking, and the tracking of an earthquake rupture along a fault line, the motion transition model $P(\theta_t|\theta_{t-1})$ can be written down as a closed form expression in terms of

θ_{t-1}. However, a natural motion model may not exist in the scenario where a radiation source is carried by a human or a vehicle traveling in an open field. In this case, it is more suitable to adopt a general motion model that assumes the event may move in any direction with equal probability in the next time step. The resulting effect of this motion model is a smoothed posterior distribution centered around the estimate of the previous time step $\hat{\theta_{t-1}}$. That is

$$P(\theta_t|\theta_{t-1}) = K(\hat{\theta_{t-1}})$$

where K is the smoothing kernel (e.g. uniform, Gaussian). The hyperparameters of the kernel depend on the speed of the target and the size of integration window, and can be adjusted dynamically. With this random motion model, the prior for next iteration becomes:

$$P(\theta_t|\mathbf{z}_{t-1}) = K(\hat{\theta_{t-1}})P(\theta_{t-1}|\mathbf{z}_{t-1})$$

Integrated Algorithm. The modified integrated algorithm uses the modified Bayesian algorithm with the prior distribution constructed from estimates produced by the modified `K-sigma` algorithm with aging. The composition of the two algorithms is the same as the stationary scenario described in Sec. 3.3.

4.1.2 Simulation Results

We simulate a 10 mCi Cesium-137 source traveling in a straight line along two trajectories at a speed of 0.5 m/s (a slow human walking speed) in a 3 x 3 sensor grid, as illustrated in Fig. 4.1. Results compiled from 100 runs for each trajectory are shown in Fig. 4.2. Results using the unmodified Bayesian algorithm for the no-motion assumption are plotted for reference. For both types of trajectory, all three modified algorithms — Bayesian with good priors, classical (`K-sigma`), and `integrated`— are able to closely track a moderately strong source traveling at this slow speed. Moderate improvement is also observed from the integrated algorithm compared to the pure parametric algorithm.

4.1.3 Testbed Results

Using the setup represented in Fig. 4.3, we conduct a set of experiments in which a 1 mCi source is moved at a constant velocity equivalent to 0.5 m/s along Trajectory A or B. In Trajectory A, the source moves down a straight path in the middle of the two array of three sensors. In Trajectory B, the source moves in a line 1 meter from one array of sensors. At $T = 0$, the source starts moving from 3.5 meters left to the grid. At approximately $T = 70$, the source reaches the end of the grid and terminates the experiment. For each trajectory, 5 runs of 70 seconds each are recorded.

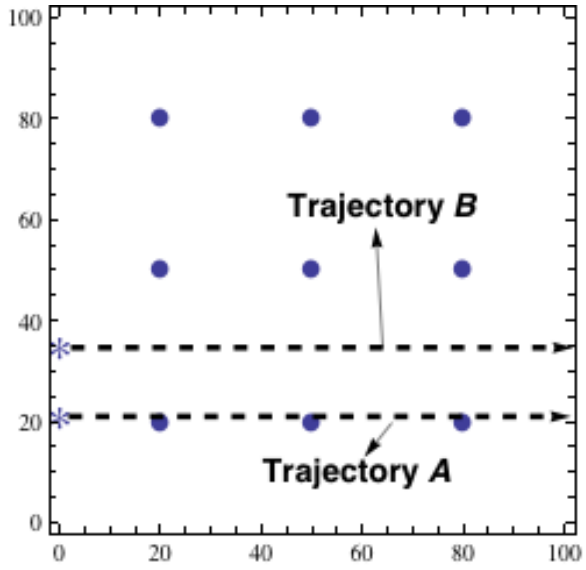
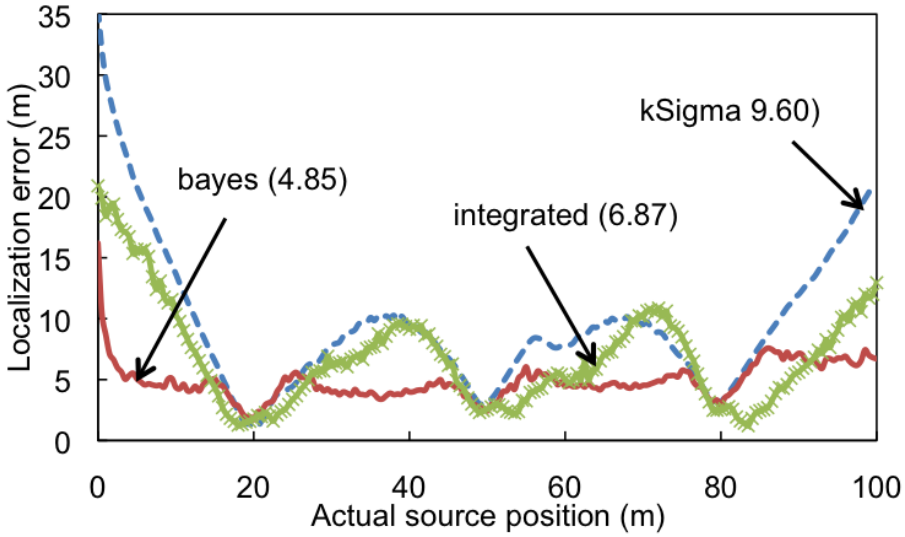


Figure 4.1: Simulation experiment setup. The circles mark the positions of the sensors. A source follows either the two trajectories across the sensor grid at a constant speed of 0.5 m/s.

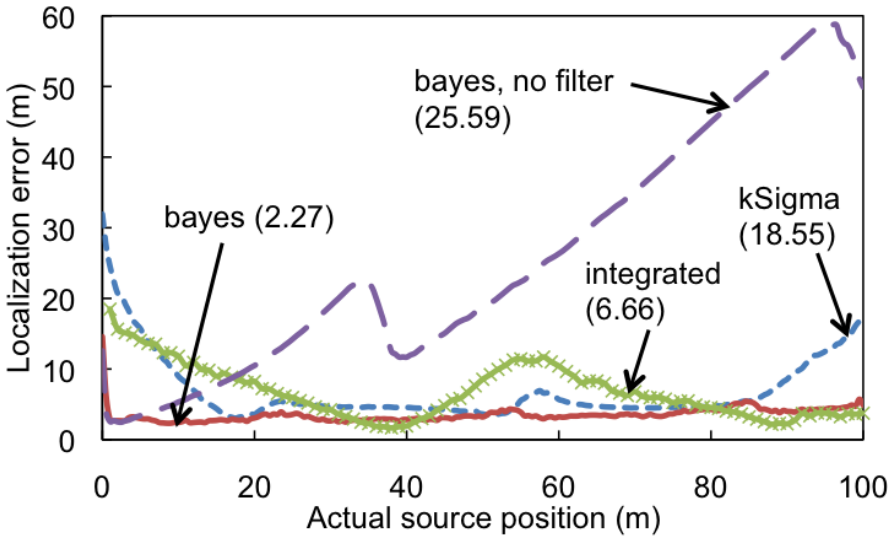
Fig. 4.4 shows 3 snapshots of the posterior distribution of the integrated algorithm during one of the runs along Trajectory B. The aggregated results for the two trajectories are shown in Fig. 4.5. The horizontal axis marks the actual source location. The vertical axis shows the localization errors at that location. For Trajectory A, three local minima at 0, 15, 30, occur when the source passes through those corresponding sensor locations at 1 meter away. For Trajectory B, the fluctuations are smaller because the source is further away from all sensors. In both setups, all three algorithms demonstrate good capability of tracking a considerably weak 1mCi source. These results correspond closely to those generated in simulation, and thus validate the feasibility of the algorithms in target tracking. No clear comparison can be made among the three algorithms though, because of the small number of trials (5 runs for each trajectory).

4.2 Mobile Sensors

Detecting a weak event with noisy sensors in a cluttered environment such as a city is very challenging. In the past, the problem has been studied in simulated scenarios where a few taxis equipped with sensors are passively detecting a radiation source inside a city [20, 41]. In this study, we present an active detection approach using sensors mounted in police patrol cars. Although the results cannot be validated with actual deployment, the simulation is based on an extensive set of real measurements collected in a city.



(a) Trajectory A



(b) Trajectory B.

Figure 4.2: Simulated source tracking of a 10 mCi Cesium-137 source traveling at 0.5 m/s. The averaged localization error over the total 200 seconds for all 100 runs is included in the parentheses next to each algorithm label. All three modified algorithms are able to closely track a moving source. The `integrated` algorithm shows moderate improvement over the classical method (`K-sigma`)

4.2.1 Local Fusion Detection Model

In the open field detection scenario, data from all sensors are considered together to make a detection decision. Combining data from multiple sensors ("sensor data fusion") improves the system-wide detection performance as measured in terms of ROC curves. However, when the effect of the event decreases too rapidly with distance, the improvement is limited [58][73], especially when the environment is cluttered with objects that occlude the source from the sensors. When devising a system

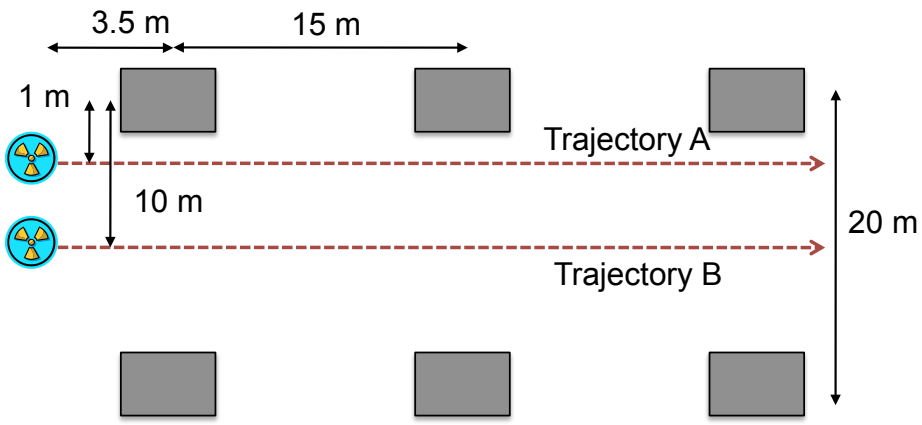
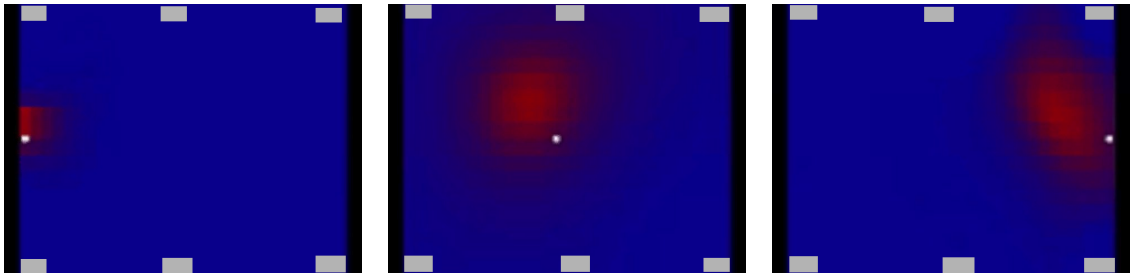


Figure 4.3: Laboratory testbed schematic, showing the six radiation sensors (the grey boxes) and the three source positions.



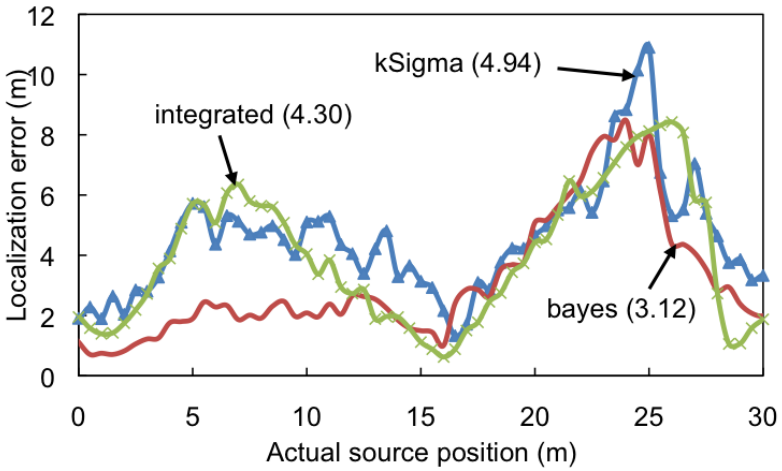
(a) T = 0 (b) T = 35 (c) T = 70

Figure 4.4: Snapshots of the posterior distribution when running the integrated algorithm on one of the runs. White dot: source, gray boxes: sensors, red: high source probability, blue: low source probability.

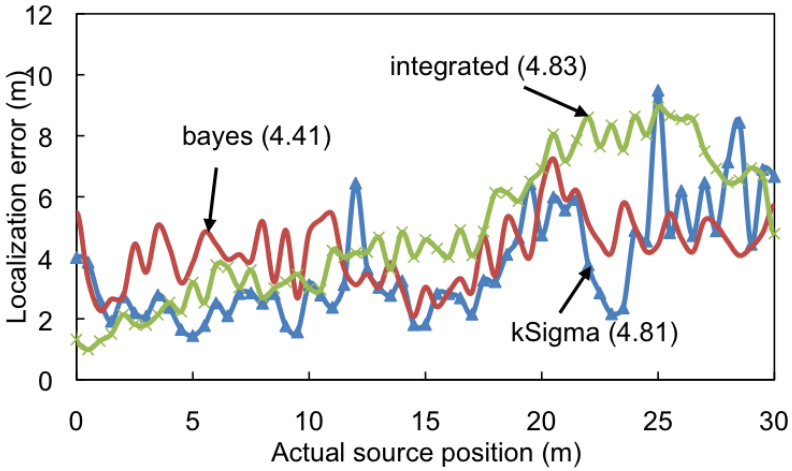
for detecting a source on the streets in an urban area, the cost of fusion may exceed the benefit of it. A different data processing model is therefore required, where only data from sensors in a relatively small area is fused, e.g. `K-sigma` algorithm with dynamic grouping as introduced in Sec. 3.1.1. In this study, we consider a scenario in which N patrol cars are seeking a radiation source placed at an unknown location somewhere on one of the city streets. For simplicity, we focus on a local anomaly detection scheme.

4.2.2 Sacramento City Radiation Map

To create a simulation environment as accurate as possible to reality, we acquire a large set of background radiation data in the downtown Sacramento city that was collected by scientists from Lawrence Livermore National Laboratory. This data set consists of measurements made over a 5-day period of $\sim$ 6 hours on each day, measured by two large NaI scintillators placed in the rear of a van. Each sensor measures 2x4x16 cm^3 with an energy resolution of 6%. The van's GPS position,



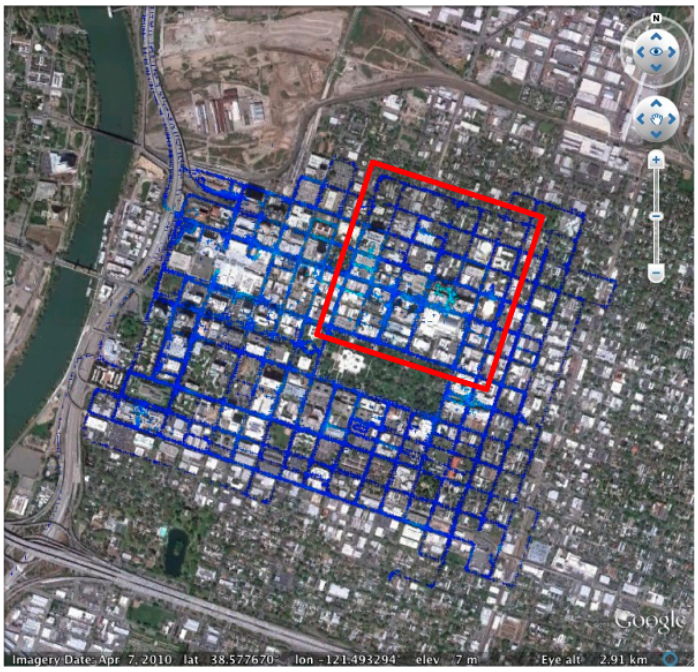
(a) Trajectory A.



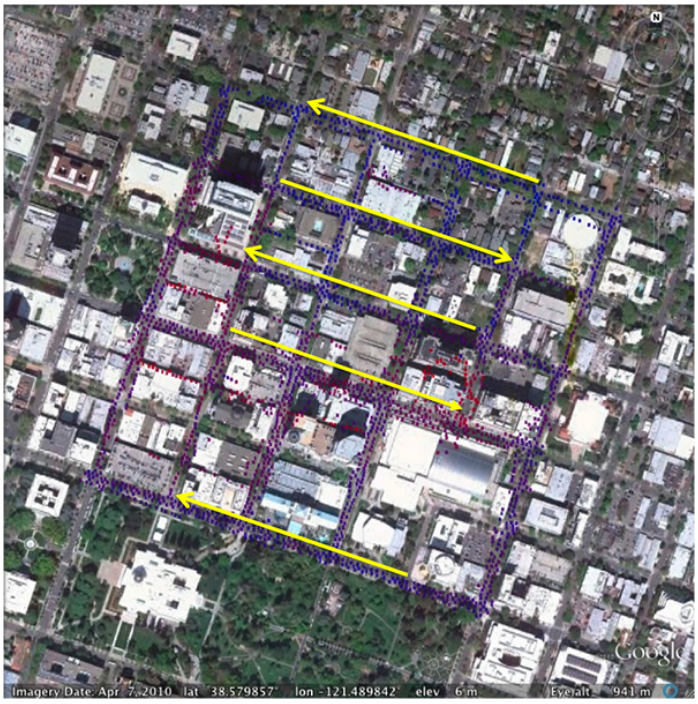
(b) Trajectory B.

Figure 4.5: Testbed source tracking of a 1mCi Cesium-137 source traveling at 0.5 m/s along two horizontal trajectories. The averaged position error over the total 70 seconds run for all five runs is included in the parentheses next to each algorithm label. All three algorithms are able to closely track a weak source as the simulations predict.

together with spectrum data, was recorded at one second intervals. The data covers about a 12x12 blocks area with multiple visits for each point (at least 5 times) at different time of the day. The data from the two sensors were combined, cleaned and corrected for calibration errors. For our study, an area of 5x5 blocks within the data set is selected. Inside this region, there are obvious fluctuations of background noise, ranging from a minimum of 500 cps to a maximum of 1200 cps as visualized in Fig. 4.6.



(a) Background radiation measurements from the compelte data set



(b) Background radiation measurements in the 5 x 5 blocks are used in the study

Figure 4.6: Sacramento background radiation data visualized in Google Earth. (a) Full set of 30 hour data. Lighter blue indicates higher noise. The sub-region used in this study is circled in red. (b) The 5 x 5 blocks area that is used in the simulation study. All East-West streets are one-way as marked by the arrows. All North-South streets are two-way

4.2.3 Simulated Detection in a City

We simulate a set of sensors that travel the city streets at an arbitrary constant speed, respecting the one-way system where present. For each simulation, a source of strength 1mCi is placed at random somewhere along one of the streets, and at 4 meters from the center of the street. The intent is to simulate a dirty bomb or a similar source placed, for example, in a backpack at the side of the street. For ease of analysis, we assume an unshielded, isotropic radiation emission pattern. At $T = 0$, the sensors (i.e. patrol cars) initially placed at random positions in the grid start moving at a constant speed.

The simulated counts are based on the real noise profile of the Sacramento streets at a one-meter resolution. A simple threshold on the `K-sigma` value is used to determine if the source is detected by the sensor in question, and if so, the simulation run is terminated, and a record is made of the elapsed time to detection t. We assume that if a patrol car detects the probable presence of a source, then the officers in the car will search the nearby area and detect the source, with probability 1.0, if a source is present.

When a sensor reaches a junction, a decision is made as to which street the sensor will start moving along. The decision takes into account the one-way system, and we prevent sensor from doubling back, unless that is the only option. Otherwise, the sensor's new street was assigned at random from the available (up to three) possibilities.

4.2.3.1 Optimizing Sensor Number and Speed

Fig. 4.7 shows the average elapsed time for varying number of sensors in search of a 1 mCi source. Two curves are displayed: one where the sensors move at 10 m/s (∼22mph) and the other where they move at 25 m/s (∼56mph). The improvement in detection time is marginal above half a dozen or so sensors, and the speed of detection is faster when the sensors travel at the higher speed.

There is, however, a tradeoff between speed and detection rate: the faster the sensor moves, the smaller the effective sensor integration window is [48], and the less likely that the source will be detected. This is illustrated in Fig. 4.8, which shows, for a set of eight sensors, the average detection time as a function of speed for sources of two strengths: 1 mCi and 0.1 mCi. We observe that if the sensor speed is increased from 40 to 50 m/s, the time to detect a source increases. Likewise, if the sensor speed is decreased from 20 to 10 m/s, the time to detect a source increases again. In fact, the speed that minimizes detection time in both cases is around 25 m/s (∼56mph). This observation suggests that in a situation where eight police patrol cars carrying detectors are dispatched in response to a tip-off that a dirty bomb is located somewhere in the city region, they should drive as fast as possible consistent with safety, up to a limit of 56mph, in order to detect the bomb in the shortest possible time. They should not, however, go faster than 56 mph.

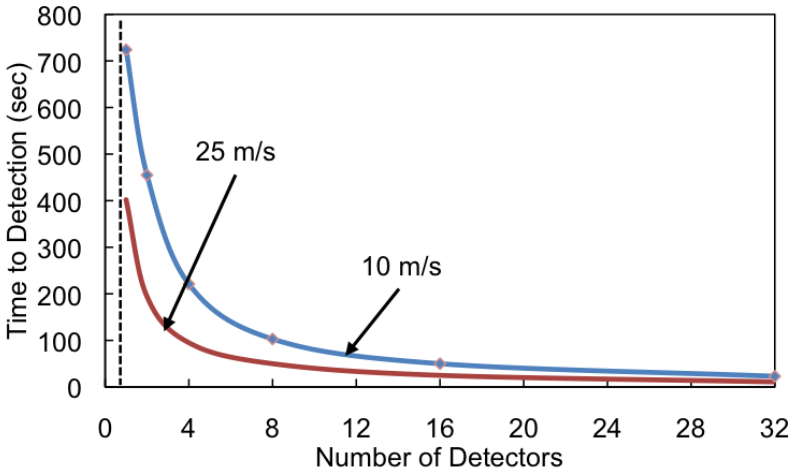


Figure 4.7: Time to detect as a function of the number of sensors at two speeds. There's only marginal improvement for $N \geq 8$.

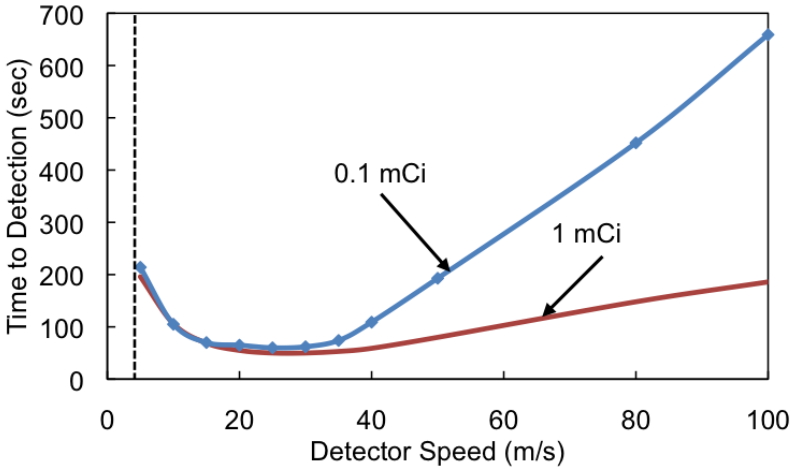


Figure 4.8: Time to detect as a function of sensor speed using two different source strengths, for N=8 sensors. There is an optimal speed at 25 m/s (56 mph) where the detection time is minimized.

In a further study, we examine the detection sensitivity as a function of the source position, i.e. whereabouts in this 5x5 block city region are dirty bombs most likely to be detected quickly. This study involved 1,000,000 separate simulation runs, each using 32 detectors traveling at 25 m/s (an optimally realistic scenario). Detection times at each of the one-meter spaced grid positions in the region are averaged to obtain a sensitivity map, shown in Fig. 4.9. In this map, the green band alongside each street (colored blue through red showing background intensity of low to high) indicates the sensitivity: the more intense the green is, the more sensitive the detection capability if a source were placed at that location.

The result in Fig. 4.9 shows that the *sensitivity* of a particular point depends predominantly on how often it is passed by, and not as much on the background level (given that the sensors are as

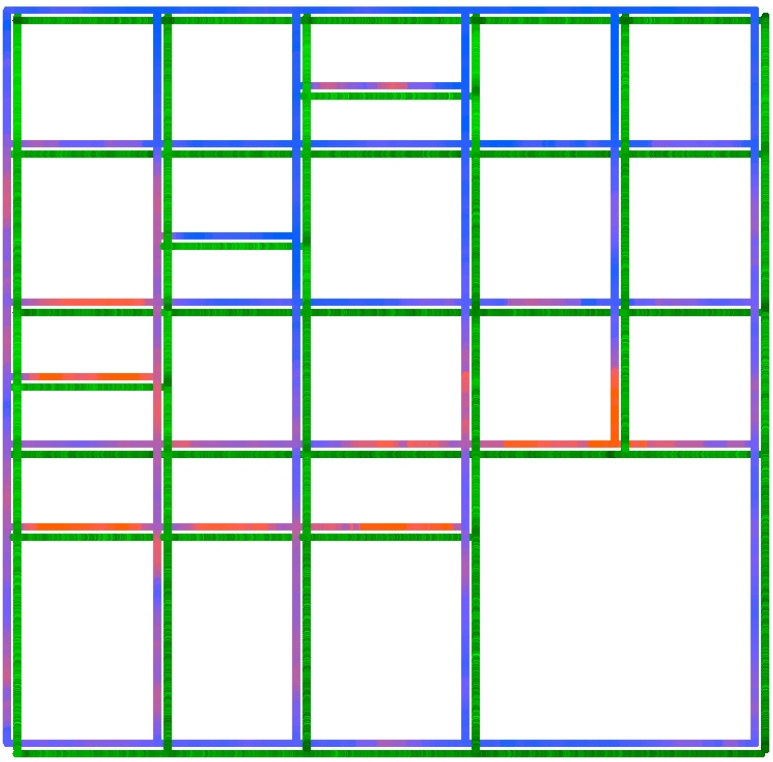


Figure 4.9: Detection sensitivity map. The level of green indicates how long it will take for the network to detect a 1mCi source placed at that point. This is shown alongside the background radiation level (blue lowest, red highest) along each street. There is no clear correlation between the background intensity and the detection sensitivity.

large as the ones in this data set). However, this may change when the target to detect is heavily shielded (weaker) or when the sensors are less sensitive.

4.2.3.2 Game Theoretic Search in a City

The detection problem essentially becomes a search problem when the source is bright enough. What then is the best strategy for patrol cars to use in order to foil a terrorist? A random traveling strategy in which all exits from a junction are taken with equal probability is sub-optimal, as it leaves some streets, such as those in the southeast corner of the city, relatively poorly covered; sensors pass there infrequently simply because there are limited routes to get to those streets. We outline below how to calculate an optimal strategy. We make the standard game theoretic assumption of rational adversaries and compute a strategy that obtains the best outcome in the worst case.

Consider a graph in which each stretch of street, without turns, is modeled as an edge. Edges are directed. A junction of streets is represented by a vertex. We want each meter of street to be traveled with the same frequency, i.e., the time interval between repeated visits for each meter segment of street should be the same. This is because if a street segment were not traveled for a

long time then a terrorist would game the system by placing a threat on that segment. Likewise, if streets were patrolled in a deterministic manner, such as first patrol East-West streets from North to South, then the terrorist could estimate the instant in the patrolling schedule with the greatest duration before the arrival of the next patrol car. Therefore, we use a probabilistic strategy in which patrol cars make turns at junctions randomly, and where the probability of a given turn is specified. A practical result of this strategy analysis will be instructions to patrol cars, for example "When you come to the junction of Avenue X and Street Y traveling East, go North 60% of the time, South 10% of the time, make a U-turn 5% of the time, and keep going East the rest of the time".

The analysis explores the flow of patrol cars on the city streets. Let $\mathit{flow}[j,k]$ be the number of patrol cars per hour that travel along street (edge) $[j,k]$, i.e., the street from junction j to junction k. For a two-way street between junctions j and k the total flow of patrol cars along the street is the sum $\mathit{flow}[j,k] + \mathit{flow}[k,j]$. For each junction j, conservation of cars gives the equation:

$$\sum_i \mathit{flow}[i,j] \;=\; \sum_k \mathit{flow}[j,k]$$

Our goal is to maximize the minimum flow amongst all the streets; this can be formulated as a linear programming problem. We use a fast heuristic to solve this problem. Fig. 4.10 shows that the heuristic increased the minimum flow rate from 0.45 cars per unit time to 0.71 cars per unit time.

4.3 Sensor Placement

Next, we study the problem of placing sensors in a fixed size field to optimize system detection performance. The problem may appear similar to the coverage and monitoring problem, but the objective is quite different: The coverage problem aims to detect an intruding target, assuming each sensor has a fixed sensing region [37]. The monitoring problem maps out the environment by optimizing mutual information [52]. The detection problem studied here optimizes the total true positive rate over the whole field. This analysis can be extended to the detection of a wide range of events.

4.3.1 Detection Function Definition

In many event detection scenarios, the ability of an individual sensor to make a correct detection decision drops as the sensor moves away from the event. This behavior can be described with a sensor detection characteristic function ϕ that is unique to the sensor type. ϕ denotes the detection probability (true positive rate) as function of distance r to the anomaly, signal strength at the sensor λ, background noise Γ, and desired sensor false positive rate f_s. In the example of radiation detection, using the Gaussian approximation for Poisson distribution, ϕ can be written down as the

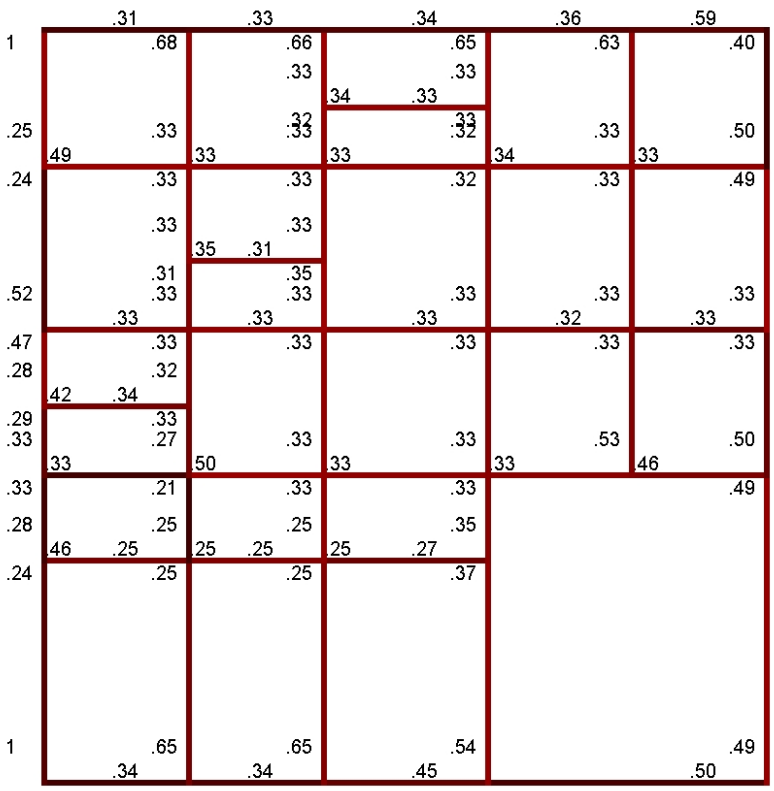


Figure 4.10: Street selection probabilities that minimize the range of traffic flows, as derived by an approximating heuristic. The numbers adjacent to each street at each junction show the probability that should be used when choosing that street when leaving the junction. With this strategy, the minimum flow is increased from 0.45 cars/unit time to 0.71 cars/unit time.

following:

$$\begin{aligned}\phi(\lambda,\Gamma) &= \text{sensor true positive rate} \\ &= \frac{1}{2} - \frac{1}{2} erf\left[\frac{Z_{TPR}}{\sqrt{2}}\right] \\ &= \frac{1}{2}\left(1 + erf\left[\frac{\sqrt{2}\lambda - 2\sqrt{\Gamma}\cdot erfc[f_s]}{2\sqrt{\Gamma+\lambda}}\right]\right) \end{aligned} \tag{4.2}$$

The second equality comes from the tail of Gaussian distribution, thresholding at Z_{TPR} (the threshold that satisfies the false positive rate requirement). $erfc$ is the complement of error function erf. λ is a function of the absolute signal strength μ, the position of the source x, and the position of the sensor s. Γ is a function of the sensor position s.

System false positive and true positive rate. We define that a system *true positive* occurs when at least one sensor detects in the presence of a target and system *false positive* as when at least one sensor detects when the target is absent. The system false positive rate f_m can be defined

in two ways:

1. per sensor: ϕ does't change with n. $\forall n$, $f_s = f_m$.

2. per map: f_s decrease as the number of sensors in the field increases since f_m stays constant for all n. f_s is adjusted as a function of the total number of sensors n.

$$f_s = 1 - \exp\left[\frac{\ln(1-f_m)}{n}\right] = 1 - (1-f_m)^{\frac{1}{n}} \tag{4.3}$$

We denote ϕ^n as the per sensor true positive rate when there are a total of n sensors in the field and define our objective detection function F as the sum of true positive rate at each possible source location x over the whole map, while keeping the system false positive rate constant.

$$\begin{aligned} F &= \text{true positive rate over the whole field} \\ &= \int_x \Big(\text{at least one sensor detects the target at x}\Big)\, \mathrm{d}x \\ &= \int_x \Big(1 - \prod_{i=1}^{n} (1 - \phi^n(\lambda(x, s_i), \Gamma(s_i)))\Big)\, \mathrm{d}x \end{aligned} \tag{4.4}$$

4.3.2 Greedy Approximate Algorithm

Computation of a set of sensor location $s = \{s_1, \ldots, s_n\}$ that maximizes Eq. (4.4) is intractable. Instead, we approximate the optimal solution using a greedy approach: the additional (n+1)th sensor is placed so as to maximize Eq. (4.4), while fixing the n sensors that have been placed already. The greedy approach is guaranteed to perform at least a fraction $(1 - 1/e)$ of the optimal solution if the target function is monotone and *submodular*, i.e. the function has diminishing returns. These conditions are formally stated in [67] as shown below:

1. Monotonicity: $A \subseteq A' \subseteq V$, $F(A') \geq F(A)$

2. Submodularity (diminishing returns): $A \subseteq A' \subseteq V$ and $y \in V \setminus A'$, $F(A \cup y) - F(A) \geq F(A' \cup y) - F(A')$

We show that the greedy bound applies to the optimization of F by showing that F is monotone and submodular.

Monotonicity. We first prove that the detection function F is monotone by showing $F(n+1) - F(n) \geq 0, \forall n \geq 0$. As a shorthand, we denote $q^n_{i,x}$ as the sensor i's false negative rate for a target at location x, $q^n_{i,x} = 1 - \phi^n(\lambda(x, s_i), \Gamma(s_i))$. With Eq. (4.4), $F(n+1) - F(n)$ can be expressed as

$$
\begin{aligned}
F(n+1) - F(n) &= \int_x \Big(1 - \prod_{i=1}^{n+1} q_{i,x}^{n+1}\Big)\,\mathrm{d}x - \int_x \Big(1 - \prod_{i=1}^{n} q_{i,x}^{n}\Big)\,\mathrm{d}x \\
&= \int_x \Big(\prod_{i=1}^{n} q_{i,x}^{n} - \prod_{i=1}^{n+1} q_{i,x}^{n+1}\Big)\,\mathrm{d}x \\
&= \int_x \prod_{i=1}^{n} q_{i,x}^{n}\Big(1 - q_{n+1,x}^{n+1} \prod_{i=1}^{n} \frac{q_{i,x}^{n+1}}{q_{i,x}^{n}}\Big)\,\mathrm{d}x \qquad (4.5)
\end{aligned}
$$

It is obvious that Eq. (4.5) ≥ 0 when using definition (1) of system false positive since $q_{i,x}^{n+1} = q_{i,x}^{n}$ and $0 \leq q_{i,x}^{n} \leq 1$. However, with definition (2) of system false positive, the inequality is less obvious. Below we sketch a proof by induction using definition (2).

It is clear that $F(1) - F(0) \geq 0$. Suppose that $F(n+1) - F(n) \geq 0$, we are required to show that $F(n+2) - F(n+1) \geq 0$ to complete the proof.

$$
F(n+2) - F(n+1) = \int_x \prod_{i=1}^{n+1} q_{i,x}^{n+1}\Big(1 - q_{n+2,x}^{n+2} \prod_{i=1}^{n+1} \frac{q_{i,x}^{n+2}}{q_{i,x}^{n+1}}\Big)\,\mathrm{d}x \qquad (4.6)
$$

Observe that $q_{i,x}^{n}$ is monotonically increasing with respect to n since $q'(n) \geq 0$ and $\frac{q_{i,x}^{n+1}}{q_{i,x}^{n}} > \frac{q_{i,x}^{n+2}}{q_{i,x}^{n+1}}$ since $q''(n) \leq 0$ and $0 \leq q(n) \leq 1$. It is not difficult to show that $q_{i,x}^{n+2}\left(\frac{q_{i,x}^{n+2}}{q_{i,x}^{n+1}}\right)^{n+1} \geq q_{i,x}^{n+1}\left(\frac{q_{i,x}^{n+1}}{q_{i,x}^{n}}\right)^{n}$ by taking the logarithm of both sides and expanding the terms. This means that the term inside the parentheses in Eq. (4.6) is greater than that in Eq. (4.5). Given this and $q_{i,x}^{n+2}\left(\frac{q_{i,x}^{n+2}}{q_{i,x}^{n+1}}\right)^{n+1} \leq 1$ and $F(n+1) - F(n) \geq 0$, we have $F(n+2) - F(n+1) \geq 0$. This proves that F is monotone.

Submodularity. Next we show that F is submodular by demonstrating that $F(n+2) - F(n+1) \geq F(n+1) - F(n)$. Following the proof of monotonicity, we know that $\left(1 - q_{n+1,x}^{n+1} \prod_{i=1}^{n} \frac{q_{i,x}^{n+1}}{q_{i,x}^{n}}\right)$ increases with n, so the marginal gain can be increasing. However, the ratio of the multiplier in Eq. (4.5) and Eq. (4.6) $\frac{q_{i,x}^{n+1}}{q_{i,x}^{n}}$ decreases faster with n (can be shown by taking the derivative of both sides and expand the terms), so that the product of the two terms decreases with n. Hence we have $F(n+2) - F(n+1) \geq F(n+1) - F(n)$.

With the proof of monotonicity and submodularity, we show that the greedy approach is guaranteed to produce a solution at least 63% as good as the optimal solution.

4.3.3 Results

The greedy approach on sensor placement for detection is validated with a simulated experiment using radiation detection as an example. With the detection function F in Eq. (4.4) as the objective function, we compute the placement of nine sensors in a field of 100x100 m^2 with a desired false positive rate $f_m = 0.01$. The source used in the calculation is a 1 mCi Cesium-137 with an equivalent

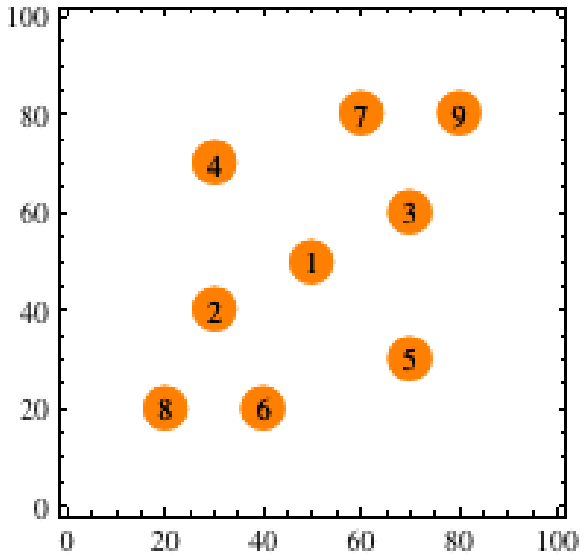


Figure 4.11: Greedy placement of nine sensors in a 100 x 100 m^2 field of uniform background. The greedy placement is not uniform.

source strength $\mu = 200$ counts per second (cps) measured by a sensor at 1 meter distance. The expected background is uniform across the field at 8 cps. The detection decision is made 60 seconds after a source is randomly introduced to the field. We restrict the minimum distance between two sensors to 10 meters and there are $\binom{11^2=121}{N}$ possible ways to place N sensors.

Non-uniform placement. The placement result for nine sensors is shown in Fig. 4.11. The marginal return for each additional sensor for up to 50 sensors is also plotted in Fig. 4.12. Note that this configuration, perhaps surprisingly, is not uniform. This points out the major difference between optimizing for detection and for coverage. The result is further supported by recent work by Malik *et al.* , who prove via Bayesian N-ary hypothesis testing that, under certain conditions, uniform sensor placement is suboptimal in maximizing detection performance [62] in terms of ROC performance.

The performance of this layout (Fig. 4.11) is evaluated in simulations with the Bayesian algorithm as described in Sec. 3.2.1, assuming that the prior on source strength is perfectly known. The resulting ROC curve is compared to those acquired from a simple grid layout (Fig. 3.7), and to a layout computed using entropy as the objective function [58]. As shown in Fig. 4.13, the layout in Fig. 4.11 performs considerably better than the grid layout. There is also a slight improvement in comparison to previous results that used entropy as objective function [58].

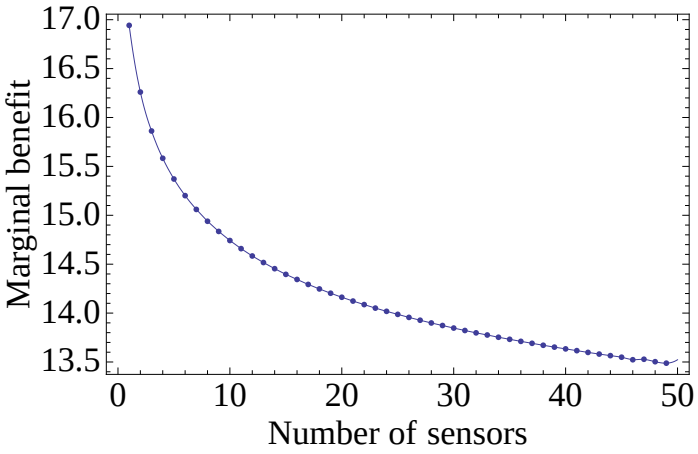


Figure 4.12: Marginal benefit for each additional sensor that demonstrates the submodular property.

4.3.4 Long Observation Time and Varying Background

The sensor's capability to make a correct decision (ϕ) improves as T, the amount of time allowed before the decision is made. The objective function F therefore depends on T as well. Fig. 4.14 shows two results when we compute the placement using different sets of prior information. In Fig. 4.14(a), the amount of time allowed for detection has been increased from 60 to 600 seconds. The placement computed approaches a uniform grid layout. In Fig. 4.14(b), we introduced a non-uniform background, modeled as a bivariate Gaussian distribution centered at coordinates (30,40), where the highest expected count was three times (25 cps) that of the lowest count (8 cps). The placement of sensors, taking into account this prior information, first covers the area with lower noise variation before placing sensors in the noisier area. This result is not surprising, as the marginal benefit of placing a sensor in an area of low SNR area is smaller than placing it in an area with high SNR.

4.4 Discussion

Event detection in sensor networks is often treated as a static problem, i.e., both the sensors and the events are stationary. In this chapter, we present three scenarios where dynamics are introduced into the network, i.e., when the event/target is moving, when the sensors are moving, and when the sensors are placed intelligently to maximize detection rate in a field. The results reported in this chapter can be used to determine the optimum movement of mobile sensors to reduce detection time.

The sensor characteristic function ϕ has the advantage that it is general enough to be modified and applied to the detection of a wide range of geospatial events, in addition to that of detecting radiation sources. However, this general approach has drawbacks compared with algorithms designed

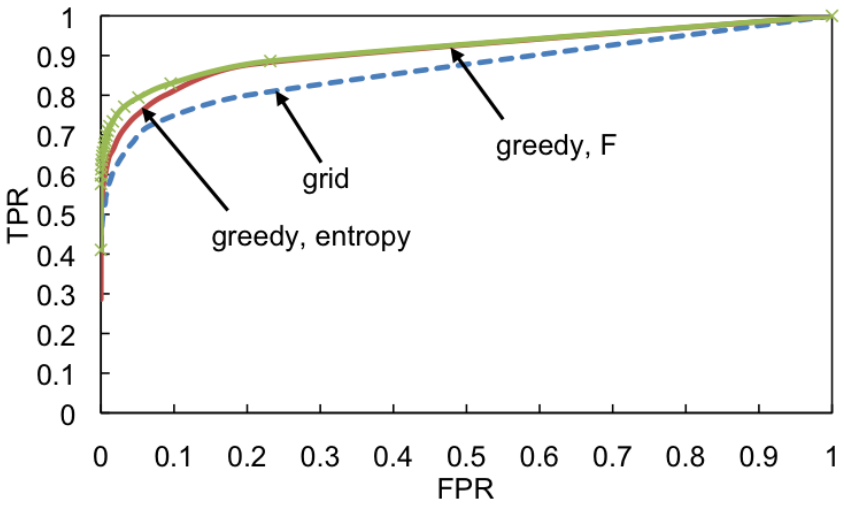
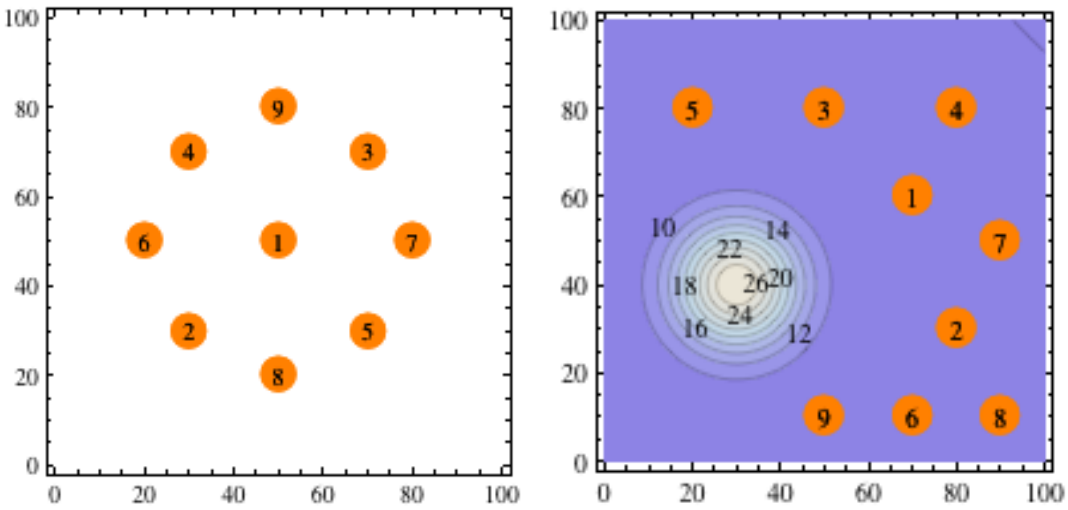


Figure 4.13: ROC comparison of three configurations of 9 sensors for detection of a 1mCi source in T=60 seconds. The greedy approach using F in Eq. (4.4) as objective function outperforms the grid baseline and the baseline using entropy as objective.



(a) T=600 (b) Non-uniform background,

Figure 4.14: Different layouts are computed as being optimal when the prior information changes. (a) Longer observation time. The sensor layout approaches uniform grid. (b) Non-uniform background modeled as a bivariate Gaussian distribution. The sensors are first placed in high SNR area.

specifically for detecting radiation sources. For example, ϕ depends on the allowable observation time T and the event intensity μ, which may not be known a priori. Though the placement results are targeted for detection, it is possible that the study can be extended to study optimal placement for parameter estimation as well.

Chapter 5

Parameter Estimation with Multiple Concurrent Events

Chapter 3 described algorithms for estimating parameters of geospatial events when the impact of at most one event is manifested at a time. Estimating parameters is more difficult when multiple events can unfold at the same time. The impact at a given location may be due to a single event or due to multiple events; distinguishing the impact at that location from different events is difficult. In fact, many existing parameter estimation methods fail in this case due to the assumption that there is at most one event/target occurring at a time. We illustrate algorithms for this problem in the context of two geospatial applications — radiation detection and earthquakes.

Radiation. NORM (naturally occurring radioactive material) sources of radiation are abundant in the environment. NORM sources include cat litter, bricks, certain rocks and minerals, and medical patients undergoing radiation treatment. Most NORM sources can be distinguished from harmful ones with spectrometers; however, most inexpensive geiger counters do not distinguish them accurately. To detect an actual threat with simple geiger counters in an environment rich with NORM sources, the algorithm needs to be able to identify the presence of multiple sources and estimate the intensity of each. An example of this problem is illustrated in Fig. 5.1, which plots the background radiation level within a 5 x 5 blocks area in downtown Sacramento, CA. Each dot is a reading from a geiger counter that repeatedly visited the area. Red and blue colors signify high and low radiation readings respectively. From the plot, at least five hot spots can be identified, which, in turn, correspond to five NORM sources.

Earthquake. After a major earthquake, multiple earthquakes can occur almost concurrently at different locations. In this case, the surface waves measured by the ground sensors can be contaminated by more than one source. If the detection algorithm assumes only one event, the event intensity can be overestimated, which in turn leads to possible false warnings. This problem is nicely illustrated with the analysis following the M9.0 Tohoku earthquake on March 11th, 2011

Figure 5.1: Raw radiation measurements collected in downtown Sacramento. Red indicates high radiation level and blue otherwise. There are multiple hot spots in the 5 x 5 blocks area, which may correspond to five NORM sources. Ignoring these potential sources may lead to poor estimate of an actual threat.

in Japan. The main earthquake was followed by a large number of aftershocks, which resulted in 70 early warnings generated in the first month after the main shock [1]. Among these, 63% of the broadcasted warnings contained significant errors, such that the estimated magnitudes were at least two orders of magnitude larger than the true value. Later analysis showed that 73% of the errors could be attributed to the problem of poor parameter estimation for multiple concurrent events [1]. Fig. 5.2 shows an example of such an error recorded on March 15, 2011, just 4 days after the main shock. Each dot represents one of the ∼1,000 stations in the Japan seismic network. Blue signifies a low shaking level and red signifies large shaking. In this instance, roughly five simultaneous events can be identified.

Contributions. This chapter presents a Bayesian algorithm for estimating the parameters of concurrent geospatial events, implemented using particle filters. As particle filter approaches are often computationally expensive, and warning applications for geospatial problems require rapid results, this chapter presents a new heuristic for speeding-up particle filters. The heuristic is validated against measurements made by the seismic network in Japan.

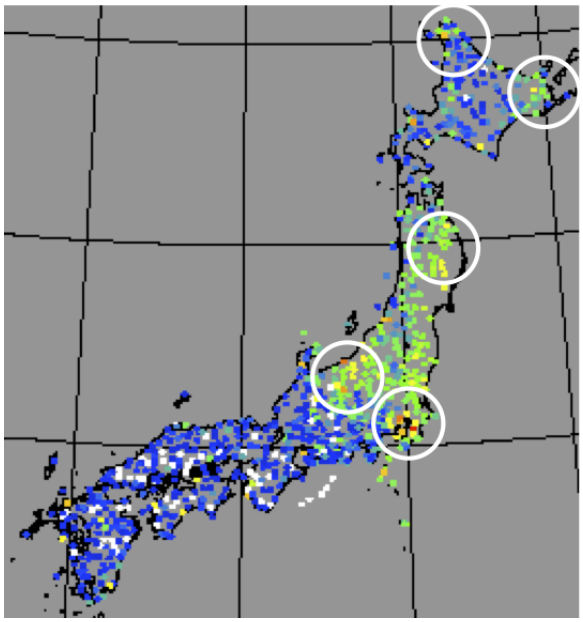


Figure 5.2: An instance of network activity recorded by the Japan seismic network on March 15, 2011. Blue and red signifies low and strong shakings, respectively. About five simultaneous events of different magnitudes can be identified from this instance.

5.1 Bayesian Formulation and MLE

The problem of continuous parameter estimation for multiple events has a natural Bayesian formulation. Let θ be the vector of parameters that characterizes an event, and Θ be a set of events that is parametrized by θ's, $\Theta = \{\emptyset, \{\theta_1\}, \ldots, \{\theta_1, \theta_2, \ldots\}\}$. Suppose $\mathbf{z}_{1:t}$ is the complete history of observations from all the sensors till the current time t; the posterior $P(\Theta_t|\mathbf{z}_{1:t})$ reveals the distribution of belief of current ongoing events at time t given the evidence and prior belief.

$$P(\Theta_t|\mathbf{z}_{1:t}) = \frac{P(\mathbf{z}_t|\Theta_t)\, P(\Theta_t|\mathbf{z}_{1:t-1})}{P(\mathbf{z}_t|\mathbf{z}_{1:t-1})} \tag{5.1}$$

where $P(\mathbf{z}_t|\Theta_t)$ is the likelihood function and is typically denoted as L, $L(\mathbf{z}_t|\Theta_t) = P(\mathbf{z}_t|\Theta_t)$. $P(\Theta_t|\mathbf{z}_{1:t-1})$ is the updated prior at time t.

$$P(\Theta_t|\mathbf{z}_{1:t-1}) = \int P(\Theta_t|\Theta_{t-1}) P(\Theta_{t-1}|\mathbf{z}_{1:t-1})\, \mathrm{d}\Theta_{t-1}$$

$P(\Theta_0|\mathbf{z}_0) \equiv P(\Theta_0)$ is the prior distribution of Θ.

Under linear motion and Gaussian noise assumption, Eq. (5.1) can be solved optimally with a *Kalman Filter*, as the posterior can be parametrized fully with a mean and covariance. Unfortunately, in most applications, including radiation and earthquake detection, Eq. (5.1) does not have an analytical solution. Sec. 5.2 will discuss several suboptimal approaches presented in [5].

Another popular method for parameter estimation is the *Maximum Likelihood Estimator (MLE)*. Instead of computing the posterior distribution with full Bayesian formation, MLE searches for the $\hat{\Theta}$ that maximizes the likelihood function and thus bypasses the need to define a prior $P(\Theta)$.

$$\hat{\Theta} = \arg\max_{\Theta} P(\mathbf{z}_t|\Theta_t)$$

While MLE is simple to apply in many applications where the model is well-defined, it is susceptible to spurious noises in the measurements, which causes the estimates to jump around the parameter space between adjacent time steps. In this chapter, we focus on the Bayesian formulation not only to avoid the this problem with MLE, but also to take advantage of prior information, such as the number of events, to simplify the computation.

5.2 Particle Filter

As mentioned in Sec. 5.1, Eq. (5.1) does not, in general, have a closed-form solution. There exists several suboptimal solutions to approximate the posterior distribution [5], one of which is to partition the parameter space into cells. This approach, though simple to apply, suffers a few problems. First of all, when the parameters are continuous and not sufficiently restricted, the grid method cannot cover the complete parameter space because there can only be a finite number of cells. Secondly, the cell approach uses a predefined cell size, and as a result, it requires a large number of cells to achieve good coverage at a desired resolution.

Another solution is the Particle Filter (PF), which is a sequential Monte Carlo method that approximates the posterior distribution with a set of weighted particles [29]. As the number of particles goes to infinity, the solution from PF approaches the optimal solution. There is a rich literature on PF and its variation, including [29],[5],[60] just to name a few. The basic procedure is summarized here for reference.

Draw Sample. At the beginning of each update step, the value of each particle is drawn from an *important density* (or *proposal distribution*) function $q(\Theta_t^i|\Theta_{t-1}^i, \mathbf{z}_t)$. For $i = 1, \ldots, N$

$$\Theta_t^i \sim q(\Theta_t^i|\Theta_{t-1}^i, \mathbf{z}_t)$$

The important density is often chosen to be the *state transition prior* $P(\Theta_t^i|\Theta_{t-1}^i)$.

Weight update. PF approximates the posterior with a collection of weighted particles

$$P(\Theta_t|\mathbf{z}_{1:t}) \approx \sum_{i=1}^{N} w_t^i \cdot \delta(\Theta_t - \Theta_t^i) \tag{5.2}$$

where w_t^i is the weight for particle i at time t. The sum of total weights are normalized to 1.

$$\sum_{i=1}^{N} w_t^i = 1$$

The weights for all particles are updated as new evidence $\mathbf{z}_t$ comes in and renormalized at the end of each update.

$$w_t^i \propto w_{t-1}^i \frac{L(\mathbf{z}_t|\Theta_t^i)P(\Theta_t^i|\Theta_{t-1}^i)}{q(\Theta_t^i|\Theta_{t-1}^i, \mathbf{z}_t)}$$

where the proposal $q(\cdot)$ is the same important density used in the sampling stage. If it is chosen to be the transition prior $P(\Theta_t^i|\Theta_{t-1}^i)$, the terms cancel out in the right hand side, and the updated weights are acquired by computing the likelihood of each particle, given the new evidence.

Resampling. Because the posterior is approximated with discrete particles, the system suffers *sample degeneracy* after a few update iterations when the weight is concentrated on a very small number of particles. The decrease in weight variance determines the degree of degeneracy that can be approximated with $\widehat{N_{eff}}$ [5].

$$\widehat{N_{eff}} = \frac{1}{\sum_{i=1}^{N}(w_t^i)^2}$$

Small $\widehat{N_{eff}}$ indicates severe degeneracy, in which case resampling is required. Resampling essentially eliminates particles with negligible weight by generating a new set of N equally weighted particles according to current distribution $P(\Theta_t|\mathbf{z}_{1:t})$. There exist many methods for sampling from a discrete distribution, which we will not discuss here.

5.3 Approximate Algorithms

Particle filtering allows for solving the Bayesian inference problem when the exact inference is intractable; however, for the estimates to approach the optimal solution, the number of particles required must grow exponentially with the number of events. While it is possible to further restrict the number of events based on prior knowledge, this is still a challenging problem in terms of computational efficiency. Can we sacrifice some accuracy in exchange for a more computationally efficient approach? In this section we discuss a few heuristics within the context of two applications — radiation and earthquake.

5.3.1 Radiation

If a radiation source can be parametrized as $\theta = [x\ y\ \mu]^T$, where $[x\ y]$ is the 2-dimensional coordinate, and μ the source intensity, then Bayesian inference through particle filtering has shown success in the presence of at most one source [77]. Since NORM radiation sources are abundant in the environment,

the number of sources in an area is difficult to estimate a priori. This means that the size of the state space (and the number of particles required) is extremely large if we directly apply Bayesian inference through particle filtering.

Static network. If the sensor positions are fixed, heuristics such as the `K-sigma` method described in Sec. 3.1.1 can be used to first estimate the number of sources ($\tilde{n}$). By adopting particle filtering with varying n around $\tilde{n}$, and examining which n maximizes the posterior probability, given the measurements $\mathbf{z}$, we can estimate both the number of sources and the source parameters $\hat{n}$, that is

$$\underset{|n-\tilde{n}|<d}{\arg\max}\ P(n|\mathbf{z})$$

The posterior probability of n is

$$\mathrm{P}(n|\mathbf{z}) \propto L(\mathbf{z}|n)P_0(n)$$

where $P_0(n)$ is the prior distribution of n and $L(\mathbf{z}|n)$ is the marginal likelihood

$$L(\mathbf{z}|n) = \int L(\mathbf{z}|n,\theta)P_0(\theta|n)d\theta$$

This technique is known as *model selection*. Morelande *et al.* have come up with similar approach that shows good performance in simulation experiments for up to three sources [63].

Mobile network. Assuming that the sensors can actively search a field [77, 87] or simply move in semi-random patterns as studied in Sec. 4.2, multiple sources can be discovered in a sequential fashion, assuming at most one source at a time in each search session. This is due to a unique property of radiation. Observe that the amount of signal measured by a sensor at distance r from a source is λ and

$$\lambda = \frac{A\mu e^{-\alpha r}}{r^2}$$

where A is the sensor's sensitivity, μ is the source intensity, and α is the photon absorption rate in air. Then the source signal drops off faster than $1/r^2$. Because most ambient sources are weak in nature, only sensors close to a source will have non-negligible measurements due to the source.

As mobile sensors can move (arbitrarily) close to a source, we can get highly accurate estimates even under the assumption of at most one source. After a source is discovered, the information can be factored into the likelihood calculation in the subsequent searches for other sources. This approximate approach grossly reduces the problem to a single source parameter estimation and has demonstrated great success in simulation.

5.3.2 Earthquake

Parameter estimation for earthquakes is, in some ways, more difficult than that for radiation because of the nonlinearity in wave propagation. Fortunately, the number of near-concurrent earthquakes n at any given time is exponentially small for large n ($n > 3$). Incorporating this information into the prior probability distribution significantly reduces the size of the state space. In comparison, n can be very large for radiation detection problems.

However, even with small n the state space may still be too large for real-time computation. For example, with the simplifying assumption that an earthquake is characterized as a point source, an event can be parameterized as $\theta = [x\ y\ D\ M\ t_0]^T$, where $[x\ y\ D]^T$ is the [latitude, longitude, depth] coordinate, M is the event magnitude, and t_0 is the event starting time. This is a 5-dimensional vector as opposed to the 3-dimensional vector in the radiation application. Suppose it requires m^k samples to correctly estimate the posterior distribution for k parameters, the difference in the number of samples between the two applications will be $(m^5 - m^3)^n$. Even when m is small ($m = 5$), it requires $\approx 10^{11}$ more samples in the earthquake estimation problem for $n = 3$.

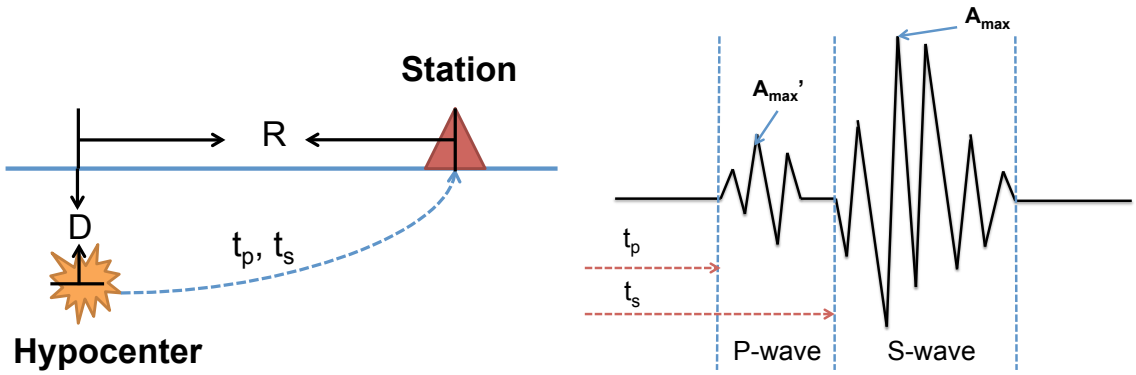
A complete Bayesian analysis may be executable in reasonable time on a parallel supercomputer; however, we present heuristics that incorporate spatiotemporal dynamics that enable particle filter computation times to grow almost linearly with n. The heuristics are validated against measurements. In the following section, we first describe a particle filter implementation for earthquakes, and then present a rapid heuristic for estimating multiple events.

5.3.2.1 Particle Filter for Earthquake Estimation

In this section we discuss the practical implementation details about designing a real-time parameter estimation system for multiple events using a particle filter. Complete pseudo code (Alg. 1) is included in the end of section.

Designing the likelihood function. If prior distribution of the parameters can be computed from historical data, the performance of particle filter for parameter estimation depends largely upon the design of the likelihood function. While this may be a straightforward process for other applications, e.g. radiation, it is nontrivial for a spatially and temporarily complex phenomenon such as an earthquake. Cua *et al.* have proposed a likelihood function formulation that assumes accurate measurements from all six channels of ground motion (maximum vertical acceleration, maximum horizontal accelerations (2-axes), velocity, and displacement) [26][24]. Here, we assume the data to be noisy and less reliable, and propose a likelihood function based on a single combined acceleration and displacement measurement.

Let A_{max} be the maximum displacement measured by a seismometer during the onset of an event; the earthquake P-wave and S-wave magnitude M_p and M_s can be expressed as a function of



(a) Hypocenter and seismic station

(b) Conceptual seismic waveform

Figure 5.3: Illustrations of the parameters used in Sec. 5.3.2.1. t_p and t_s mark the arrival time of the P-wave and S-wave since the start of the earthquake at t_0, $t_p \leq t_s$. P-wave usually has smaller top acceleration (and displacement) compared to S-wave.

R the linear surface distance to the epicenter, D the depth of the hypocenter, and A_{max}.

$$0.72M_p = \log A_{max} + 1.2 \log R + 5 \times 10^{-4} R - 5.0 \times 10^{-3} D + 0.46 \tag{5.3}$$

$$0.87M_s = \log A_{max} + \log R + 1.9 \times 10^{-3} R - 5.0 \times 10^{-3} D + 0.98 \tag{5.4}$$

The relationship between the parameters is visualized in Fig. 5.3. These formulae were specifically tailored for the geological compositions in Japan [4]. The P-wave and S-wave magnitudes are expressed in terms of the maximum displacement A_{max} rather than the maximum acceleration or velocity because the displacement metric preserves energy better over long distance. The displacement value $A(t)$ observed at time t can be calculated in real-time, which involves twice integration of the raw acceleration measurement $k(t)$ [50].

$$A(t) = gn \times [k(t) + h_0 \cdot k(t-1) + h_1 \cdot k(t-2)] - h_2 \cdot A(t-1) - h_3 \cdot A(t-2)$$

where the function gain gn and filter constants h_0, h_1, h_2, h_3 depend on the damping constant and sampling period of the seismometer. For a JMA seismometer with a 0.55 damping constant and a 6 second sampling period,

$$gn = 0.0000248691025,\ h_0 = 1.0,\ h_1 = 1.0,\ h_2 = -1.9889474,\ h_3 = 0.9895828$$

Given Eq. (5.3) and Eq. (5.4), and that the displacement is log-normally distributed $A \sim \ln \mathcal{N}(\mu, \sigma^2)$, we propose the following likelihood function for a single station.

$$L(z|x, y, D, M, t_0) = \frac{\exp \frac{-(\log A_{max} - \log A_{exp})^2}{2\sigma^2}}{A_{max} \cdot \sigma\sqrt{2\pi}} \tag{5.5}$$

Here A_{exp} is the expected A_{max} and σ is the standard deviation of displacement measurement. Depending on whether the station has observed the P-wave, the S-wave, or neither, the expected maximum displacement and its standard deviation will be different. By rearranging Eq. (5.3) and Eq. (5.4), we can compute A_{exp} and σ for the following three cases.

- Has not observed any wave:

$$\log A_{exp} = \log A_{noise},\ \sigma = \sigma_{noise}$$

- Has observed P-wave:

$$\log A_{exp} = 0.72M_p - 1.2\log R - 5 \times 10^{-4}R + 5.0 \times 10^{-3}D - 0.46,\ \sigma = \sigma_p$$

- Has observed S-wave:

$$\log A_{exp} = 0.87M_s - \log R - 1.9 \times 10^{-3}R + 5.0 \times 10^{-3}D - 0.98,\ \sigma = \sigma_s$$

A_{noise} and σ_{noise} are the noise in displacement measurement due to recent environmental noise and can be computed independently for each station by keeping a running window. σ_p and σ_s can be precomputed using historical data. R is the distance between the location of the station and the estimated epicenter.

The decision of which A_{exp} to compute for a station depends on whether the P-wave, the S-wave, or neither has arrived at the station. Using a very simple linear speed model of seismic wave one can compute the expected arrival time of the P-wave and S-wave (t_p and t_s) given the relative location of the station to hypocenter (x, y, D). More sophisticated models that take into consideration the earth's structures also exist. In this study we adopt one of these models tailored for Japan. Comparison between t_p, t_s, the current time t, and the event start time t_0 gives direct estimation of which A_{exp} to compute. Fig. 5.4 provides a illustrative summary of these design ideas.

This design of likelihood function is based on the maximum displacement A_{max} that the sensor observes when in contact with a seismic wave. However, a sensor does not observe the maximum displacement immediately after the wave arrives, but rather after a period of time. In this case, the initial estimates will be highly incorrect using this likelihood function. A delay term $\alpha(\cdot)$ can be included to approximate the instantaneous displacement before the maximum is observed.

$$A_{exp} = \alpha(t - t_0 - t_p)A_{max},\ 0 \leq \alpha(\cdot) \leq 1$$

The likelihood $L(\cdot|\cdot)$ is applied in each time step to update the weight of each particle. Assuming

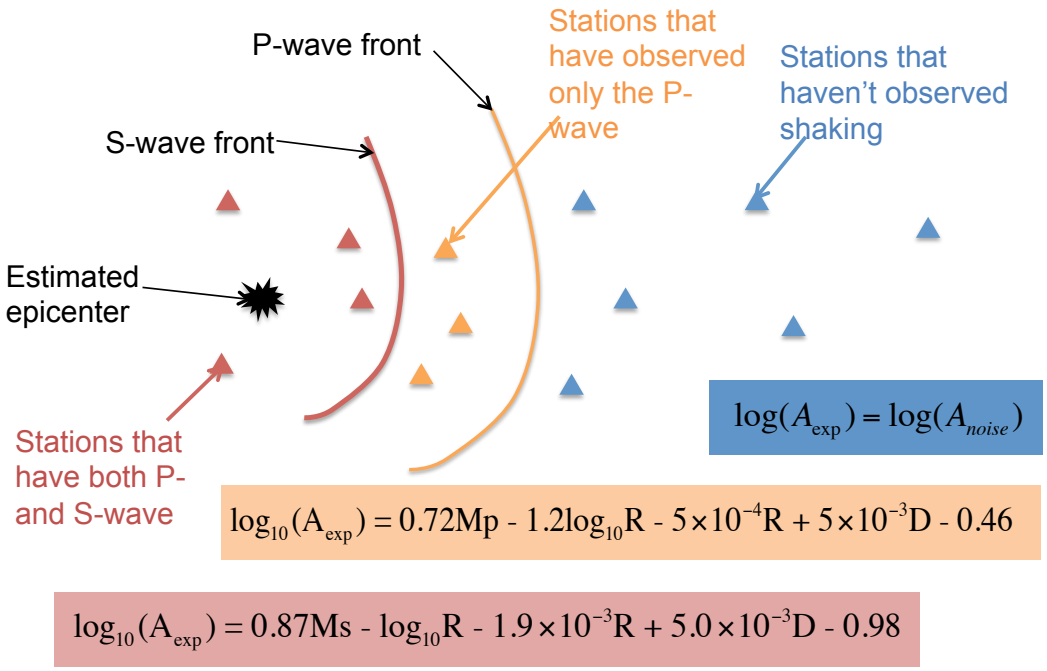


Figure 5.4: Illustrative summary of the design of a single station likelihood function. The expected observation made by a station depends on whether it should have observed the P-wave, the S-wave, or neither, given an hypocenter estimate.

that each station makes independent observations and the collection of observations from all stations is $\mathbf{z}$, the complete likelihood function becomes

$$L(\mathbf{z}|x, y, D, M, t_0) = \prod_{i=1}^{n} L(z_i|x, y, D, M, t_0) \tag{5.6}$$

where n is the number of stations. Note that the independence assumption is a minor simplification, given that nearby stations have correlated observation.

Ensuring good coverage of parameter space. Particles are initialized according to a prior distribution on the parameters. Because we are approximating an unbounded and continuous 5-dimensional space with a bounded and discrete one, care must be taken to ensure that the particles have sufficient coverage and the number of particles required stays inbound. This is especially important in the case of seismic application, as both the number of parameters and the values they can take are large. One way to ensure particle diversity with a limited number of particles is to adopt a *Regularized Particle Filter (RPF)* [5].

RPF differs from regular particle filters only in the resampling stage. Rather than sampling from a discrete approximation of the posterior density $P(\cdot|\mathbf{z})$ as in Eq. (5.2), RPF samples from a

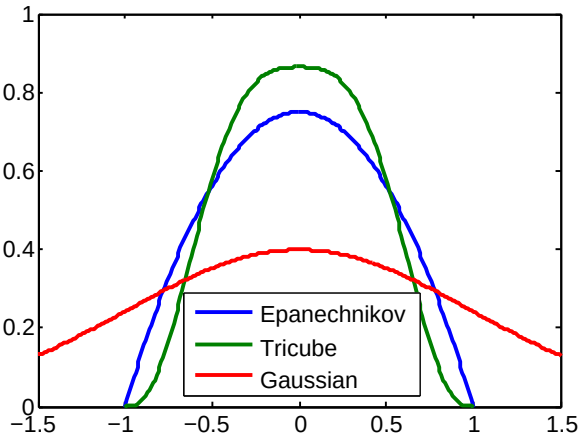


Figure 5.5: Some popular smoothing kernels used in Regular.

continuous approximation [65]. More specifically, RPF draws samples from the approximation

$$P(\theta|\mathbf{z}) \approx \sum_{i=1}^{N} w_i \cdot K_h(\theta - \theta_i)$$

where $K_h(\theta) = \frac{1}{h}K(\theta/h), h > 0$ is the rescaled kernel density of $K(\cdot)$, h is the bandwidth, and w_i is the normalized weight for particle i. As a comparison, $K_h(\theta)$ is the Dirac delta function $\delta(\theta)$ in the regular particle filter. Special care is given to the design of kernels to minimize the error between the approximated and the actual distributions. Under the assumption that all particles are equally-weighted and that its density is Gaussian, the optimal kernel is the Epanechnikov kernel [65].

$$K_{opt}(x) = \begin{cases} \frac{n_x+2}{2C_{n_x}}(1 - \|x\|) & \|x\| < 1 \\ 0 & \text{otherwise} \end{cases} \tag{5.7}$$

where n_x is the dimension of the parameter space, C_{n_x} is the volume of the unit hypersphere in $\mathbb{R}^{n_x}$. Fig. 5.5 lists a few popular kernels in the literature. The bandwidth vector h can be chosen proportionally to the variance in the particle population by computing the Cholesky decomposition of the empirical covariance matrix.

Efficient tracking of multiple events. We propose a simple heuristics to keep track of multiple events. The heuristic has the desired property that the complexity grows linearly with the number of the events.

As a first approximation, the heuristic initializes separate particle filters $pf_1(\theta_1), pf_2(\theta_2), \ldots$ for all possible events, rather than keeping track of all events within one particle filter $pf(\Theta = \{\theta_1, \theta_2, \ldots\})$. Each particle filter communicates its current estimate $\hat{\theta}$ at *the end* of each update step to all other

particle filters. Specifically, each particle filter pf_i computes the following posterior at time t

$$P(\theta_i^t|\mathbf{z}, \{\hat{\theta_j^{t-1}}, j \neq i\})$$

This approximation dramatically reduces the computations required to keep all event estimations up to date. It is suboptimal, however, because all the particles from $pf_1, pf_2, \ldots$ combined only cover a small fraction of the complete parameter space.

A new particle filter is initialized with a local detection. The threshold for local detection is set high enough such that noisy detections are filtered out. The local detection decision can be made with a simple low pass filter to identify transient changes, e.g., `K-sigma` (Sec. 3.1.1) or STA/LTA (short term average over long term average). Because local detection can be due to an existing event that is being tracked by another particle filter, it is necessary to condition new initialization on a separate metric. A natural choice of metric is $\mathbb{P}\left[z|\hat{\theta_1}, \hat{\theta_2}, \ldots\right]$, i.e., the probability that the triggered measurement is due to existing events. Computation of this metric can follow directly from the single station likelihood calculation, as in Eq. (5.5); however, determining A_{exp} is nontrivial in this case, as it involves computing the additive effect of the interference of multiple wavefronts. We propose an alternative metric that allows for rapid computation; the metric is the probability of shaking due to *any* of the existing events and threshold on the highest probability:

$$\max_i \mathbb{P}\left[z|\hat{\theta}_i\right] = \max_i L(z|\hat{\theta}_i) \begin{cases} < \tau, & \text{initialize new pf} \\ \geq \tau, & \text{do nothing} \end{cases} \tag{5.8}$$

By tuning the threshold τ, we adjust how conservative the system is in declaring new events.

Selecting prior and the number of particles. Prior $P(\theta)$ determines how the particles are initialized. A good prior encodes geographical information, such as the location of fault lines near the station that first triggered, and the most common magnitudes generated at these fault lines. This information can be compiled from historical earthquake catalogs for each station and used in real time when initializing the RPF. If prior information is absent, then a flat prior can be used. The problem of choosing the range of the prior goes hand in hand with selecting the number of particles. A prior of large coverage may cause the initial estimates to be jittery because little evidence were presented. Prior distributions that cover a small area may lead to slow convergence. We evaluate these tradeoffs empirically to select the appropriate prior coverage and the suitable number of particles.

Algorithm summary. Next we summarize this PF approach for the parameter estimation problem for multiple seismic events.

Algorithm 1: Regularized Particle Filter for multiple seismic event tracking

$PF \leftarrow \{\}$
Initialize thresholds τ, α
Initialize bandwidth vector $h' \in \mathbb{R}^{n_x}$
while *not end* **do**
  $\star$ Check for new event
  $Z \leftarrow$ list of station measurements that triggered
  for $z \in Z$ **do**
    $pr \leftarrow \max_k L(z|\theta_k)$
    if $pr < \tau$ **then**
      $\left[\{\theta_i, w_i\}_{i=1}^N\right] \leftarrow RPF\left[\{\theta_i, w_i\}_{i=1}^N, z\right]$
      for $i = 1 \rightarrow N$ **do**
        Draw $\theta_i \sim P(\theta, z)$
        Assign weights based on prior and z, $w_i \sim P(\theta, z)$
      $pf \leftarrow \left[\{\theta_i, w_i\}_{i=1}^N, z\right]$
      $PF \leftarrow PF \cup pf$
  $\star$ Update weight, resample if needed
  for $pf \in PF$ **do**
    $\{\theta_i, w_i\}_{i=1}^N \leftarrow pf$
    for $i = 1 \rightarrow N$ **do**
      $w_i \leftarrow w_i L(\mathbf{z}|\theta_i)$
    $\left[\{\theta_i, w_i\}_{i=1}^N\right] \leftarrow NORMALIZE\left[\{\theta_i, w_i\}_{i=1}^N\right]$
    Compute $\widehat{N_{eff}} \leftarrow \frac{1}{\sum_{i=1}^N w_i^2}$
    if $\widehat{N_{eff}} < \alpha$ **then**
      $\left[\{\theta_i, w_i\}_{i=1}^N\right] \leftarrow RESAMPLE\left[\{\theta_i, w_i\}_{i=1}^N, z\right]$
      for $i = 1 \rightarrow N$ **do**
        Draw $\epsilon \sim K$ from the Epanechnikov Kernel
        Compute weighted empirical covariance matrix S_k of $\{\theta_i, w_i\}_{i=1}^N$
        Compute lower triangle $D_k = chol(S_k)$, $D_k D_k^T = S_k$
        $\theta_i \leftarrow \theta_i + h' D_k \epsilon$
  $\star$ Check for termination
  for $pf \in PF$ **do**
    $\{\theta_i, w_i\}_{i=1}^N \leftarrow pf$
    if $CONVERGED\left[\{\theta_i, w_i\}_{i=1}^N\right]$ **then**
      $PF = PF - pf$

5.3.2.2 Experiments and Case Studies

We validate the approximate PF approach empirically on the Japan seismic data set. The data set contains one-month of continuous recordings from ∼200 JMA seismic stations in Japan following the Tohoku M9.0 earthquake on March 11th, 2011. The raw data is first converted to SAC format[1] and compressed down to 1 Hz. Incomplete records are discarded. Parameter estimation is carried out as described in Sec. 5.3.2.1, using a flat prior around the first triggered station and 1,000 particles for each particle filter. Since a particle filter approach has rarely been applied to earthquake parameter estimation, we first test the algorithm in the presence of a single major event. The performance of algorithms for discerning and estimating multiple events are validated on two sets of two concurrent events.

Tohoku M9.0 — Single Event. We test the performance of the particle filter approach on the Tohoku M9.0 earthquake that occurred at 14:46:18, on March 11th, 2011. The first trigger occurs at 14:46:47 at station location [38.4570, 141.3500], since which a particle filter of 1,000 particles is initialized and estimates are produced at a one-second interval for the next 120 seconds. Particles are visualized in Fig. 5.6 for the first 60 seconds since after the start. Fig. 5.6(a) shows the flat prior used to initialize the particles at 14:46:48. The estimates mostly converge and stabilize in the first 30 seconds. This is better illustrated in Fig. 5.7, which compiles the error in parameter estimates from 15 independent runs.

These results are compared against the event parameters determined by the Japan Meteorology Agency (JMA) (marked as dotted lines in Fig. 5.7) and demonstrate, on average, good performance for all parameters, although the variance between trials is not negligible in such a small number of runs.

March 15, 2011 — Two Events. One of the many false warnings in the earthquake early warning (EEW) record broadcasted by JMA happened at 1:36:26, on March 15, 2011. The warning was given based on an estimated magnitude of M5.9 (Fig. 5.8), 21 seconds after first detection. Post event analysis revealed that it was actually a M3.4 earthquake. Replay around this time shows that at least two events ∼200 km apart occur within 15 seconds of each other, which may have confused the EEW system. We ran the particle system algorithm during this period of time using 2,000 particles. Screenshots for one of the runs are shown in Fig. 5.9. The labeled times correspond to seconds since first detection of the first event. While the algorithm is able to compute and determine in real time that the two events did not originate from the same source, and converges within a couple seconds on the estimates, the variance between runs are quite large, as shown from the compiled results in Fig. 5.10.

[1]SAC format: `http://www.iris.washington.edu/software/sac/manual/file_format.html`

March 20, 2011 — Two Events. Another instance during which false warnings were issued happened at 14:20:6, on March 20th, 2011. In this case, the EEW system estimated a magnitude of M7.6, 6.6 seconds after initial detection. The magnitude was later corrected to M4.3. Again, from replays of the records, the incorrect estimate can possibly be attributed to two events ∼150 km away from each other, occurring 5 seconds apart (Fig. 5.11). Fig. 5.12 and Fig. 5.13 show that the particle filter algorithm is able to separate the two events and converges quickly in estimates for the later event (Event B as labeled in Fig. 5.12), though it has more difficulty with the first event (Event A). This is likely due to the lack of sensor density and that the event was weak and offshore.

5.4 Discussion

We discuss in this chapter a series of challenges and solutions regarding parameter estimation in the presence of multiple geospatial events. We also present a particle filter based algorithm and approximate implementation for earthquake classification. While this approach demonstrates promising performance in initial case studies, there are several areas where it can be improved.

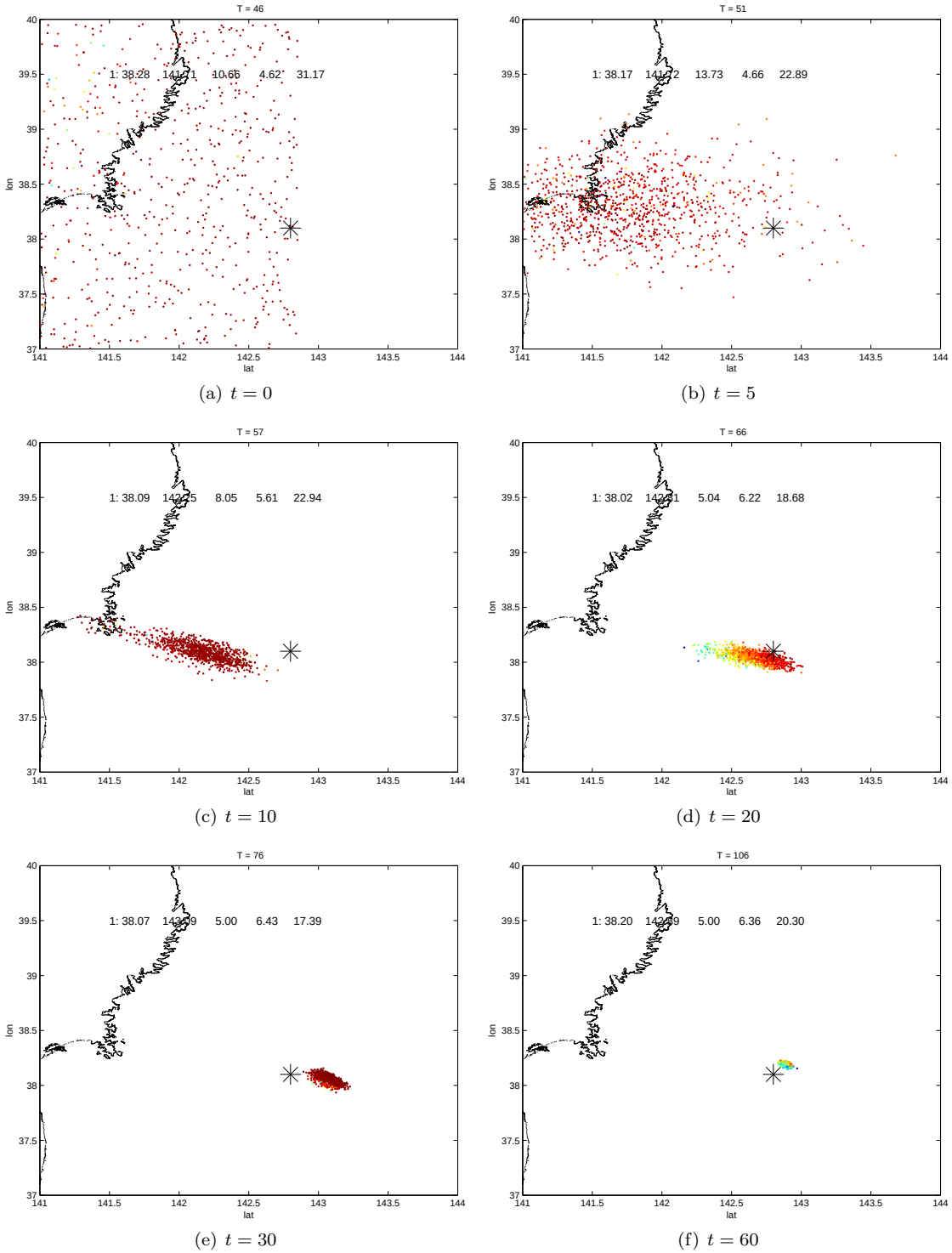
- The design of a likelihood function assumes that the sensors can be noisy and unreliable, and, as a result, utilizes only the maximum displacement information. This design, while simple, does not allow for easy incorporation of transient responses. Cua & Heaton, in [24], propose a likelihood function that utilizes six channels of information, i.e. maximum acceleration in x, y, z direction, velocity, and displacement. If high quality sensors are available, combining this with the parametrized transient ground motion, as described in [26], may lead to better estimation performance.

- The regularized particle filter approach circumvents the need for intensive computation that traditional grid search requires for poorly-defined parameter space. However, selecting a good prior and choosing a suitable number of particles are crucial for the performance. Some of the slow convergence and high variance results in Sec. 5.3.2.2 may be attributed to lesser choices in these parameters. As mentioned in Sec. 5.3.2.1, prior distribution can be acquired from historical record. Alternatively, initial measurements can be used to construct such a prior as with the integrated algorithm introduced in Chapter 3.

- The performance of parameter estimation for multiple seismic events is limited by how well one can model the ground motion when multiple wavefronts collide. In the algorithm proposed in Sec. 5.3.2.1, this problem is ignored. While the omission makes little difference in the case studies where the events are far enough apart ($>$ 100km), if we want to apply the same technique to separate from the mainshock aftershocks that are close in time, then such a model must be considered.

- The approximate method is proposed with computational efficiency in mind, such that it scales linearly with the number of events. However, for events very close together in time and space, the performance may suffer, as only single point estimates from other events are used in the update iteration. Given that the probability is extremely small for having more than two events of independent sources occurring close in time and space (not including aftershocks), it may be feasible to use a full scale particle filter with a maximum event number of two.

- While experiments in Sec. 5.3.2.2 utilize only ∼180 JMA stations in Japan, there are another ∼800 small-motion stations managed by the National Research Institute for Earth Science and Disaster (NIED). These stations are less robust than the JMA ones, but are much greater in number and coverage. As an example, Fig. 5.14 shows instantaneous accelerations measured by the JMA stations and JMA+NIED stations combined. It is clear from the acceleration plot that the combined network gives much more information and resolution about an event. It can be expected that by including heterogenous stations in the computation, the quality of estimates and the convergence rate can be improved. In addition, the integration is straightforward with the Bayesian framework. We will study the problem of retrieving useful information from a large number of lesser quality sensors in Chapter 6.

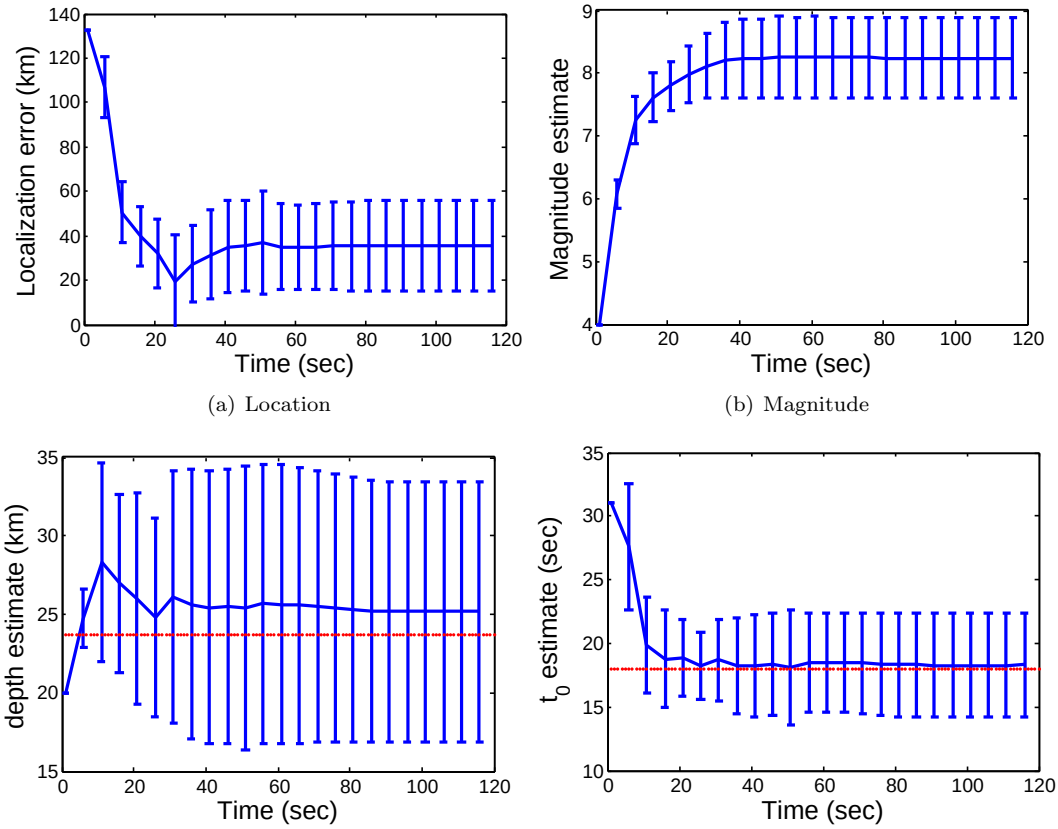


(a) $t = 0$ (b) $t = 5$

(c) $t = 10$ (d) $t = 20$

(e) $t = 30$ (f) $t = 60$

Figure 5.6: Series of visualizations of 1,000 particles for single event estimation. Red and blue colors represent high and low probability. The hypocenter determined by JMA is marked as a black cross.



(a) Location (b) Magnitude

(c) Depth (d) Start time (t_0)

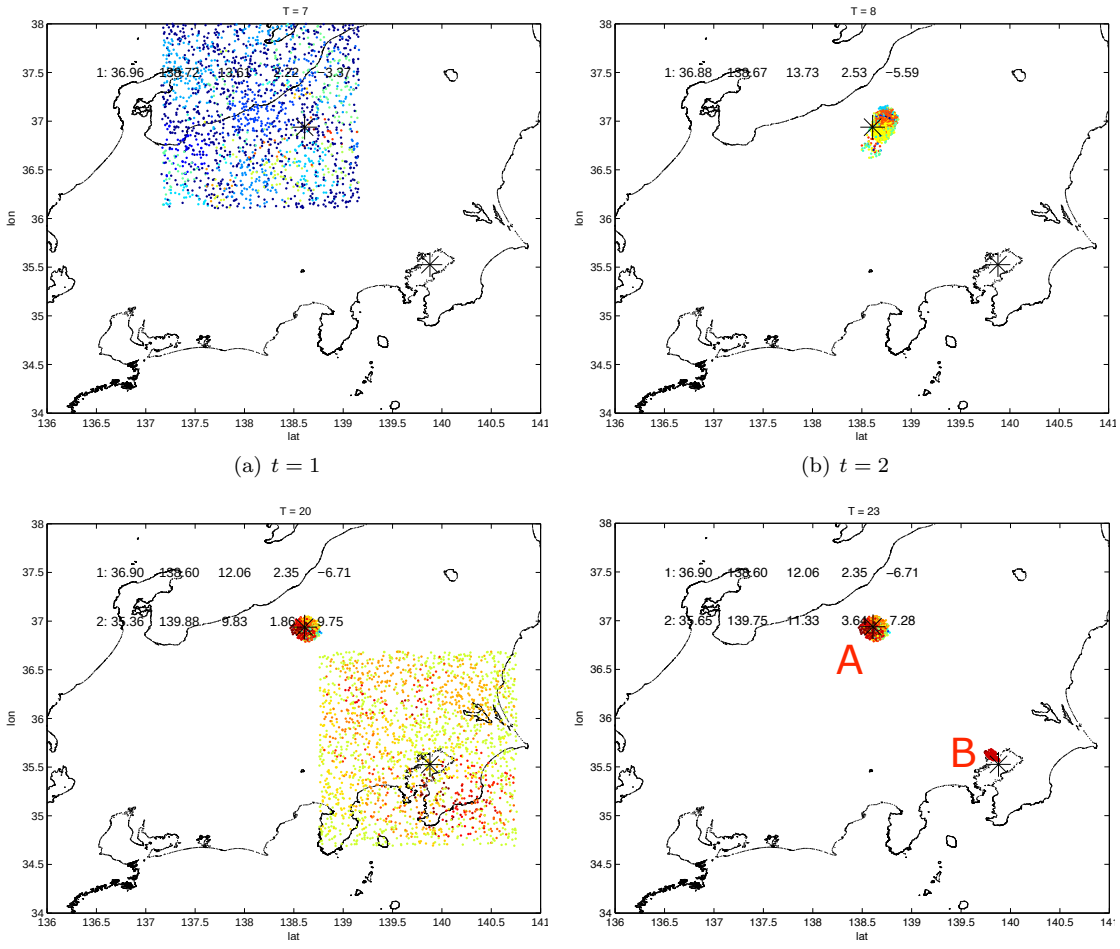
Figure 5.7: Tohoku M9.0 earthquake. Quality of parameter estimates as a function of time since the event is detected by the first station. The results are compiled from 15 independent runs and plotted using standard deviation as the error bar. The exact event parameters determined by JMA are [latitude longitude depth magnitude t_0] = [38.1 142.8 23.7 9.0 18.0]. These values are shown as red dotted lines in the plots. The estimates from the PF approach compare well with the official record.

(a) EEW location estimates

提供時刻等		震源要素等					
		Seconds elapsed since first detection	震源要素				予測震度
			北緯	東経	深さ	Magnitude	
地震波検知時刻	01時36分05.3秒	–	–	–	–	–	–
1	01時36分26.0秒	20.7	36.9	138.6	10km	5.9	※1
2	01時36分33.2秒	27.9	36.9	138.6	10km	5.9	※2
3	01時36分35.2秒	29.9	36.9	138.6	10km	5.9	※3
4	01時36分43.9秒	38.6	36.9	138.6	10km	5.3	※4
5	01時37分03.2秒	57.9	36.9	138.6	10km	5.3	※5
6	01時37分17.3秒	72.0	36.9	138.6	10km	5.3	※6

(b) Real-time EEW record

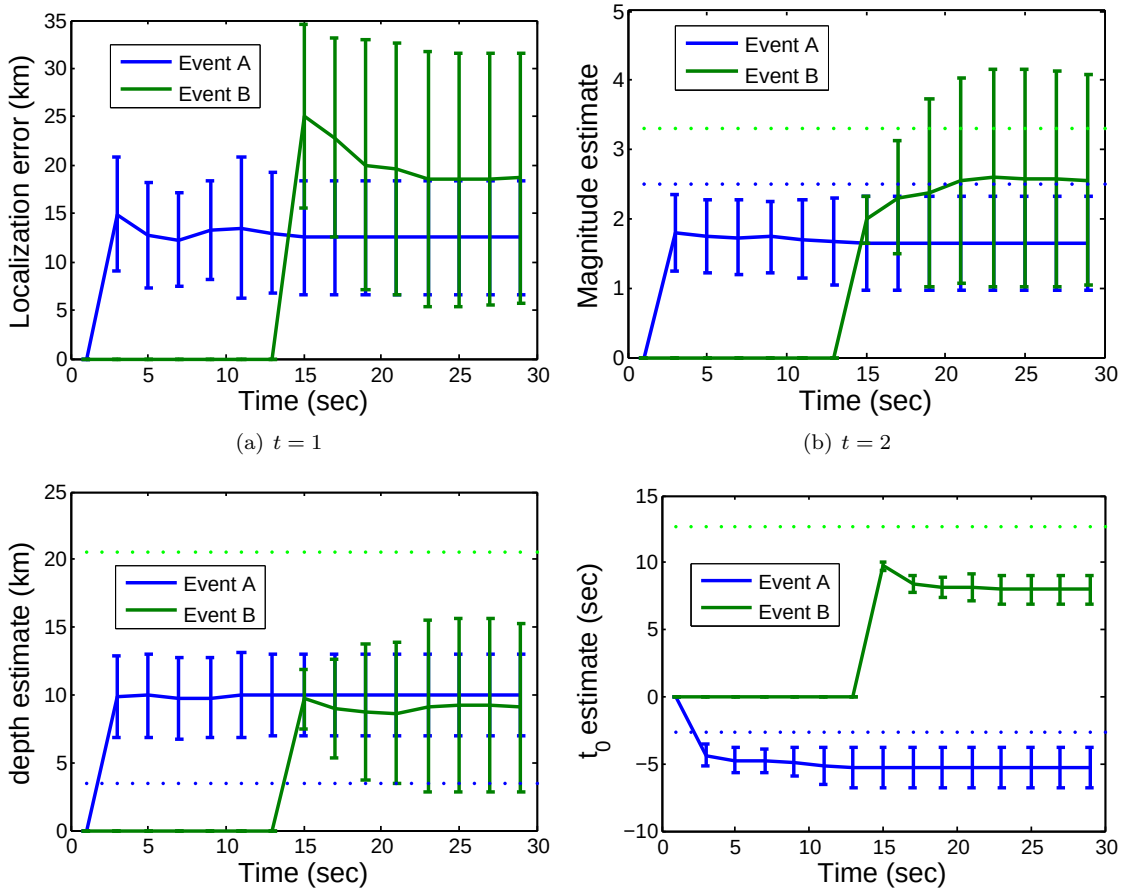
Figure 5.8: Real time warnings issued by JMA at 1:36:05, on March 15, 2011. The magnitude was over estimated by an order of 3, most likely because the two smaller events 14 seconds apart were associated together. [Source: `http://www.seisvol.kishou.go.jp/eq/EEW/kaisetsu/joho/20110315013605/content/content_out.html`].



(a) $t = 1$ (b) $t = 2$

(c) $t = 14$ (d) $t = 17$

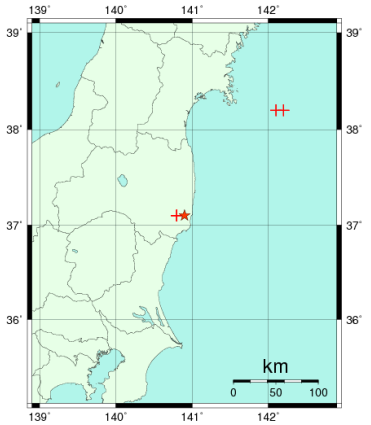
Figure 5.9: 2,000 Particles visualized between 1:36:05 and 1:36:35 on March 15th, 2011. The labeled time correspond to seconds elapsed since first detection. The two events are labeled in (d) for references. Both events converge within a few seconds since first detections.



(a) $t = 1$ (b) $t = 2$

(c) $t = 14$ (d) $t = 17$

Figure 5.10: Results compiled from 20 independent runs for the period 1:36:05-1:36:35 on March 15th, 2011. Events are labeled according to Fig. 5.9. Location estimates are within a reasonable error range; however, the magnitudes are generally underestimated.

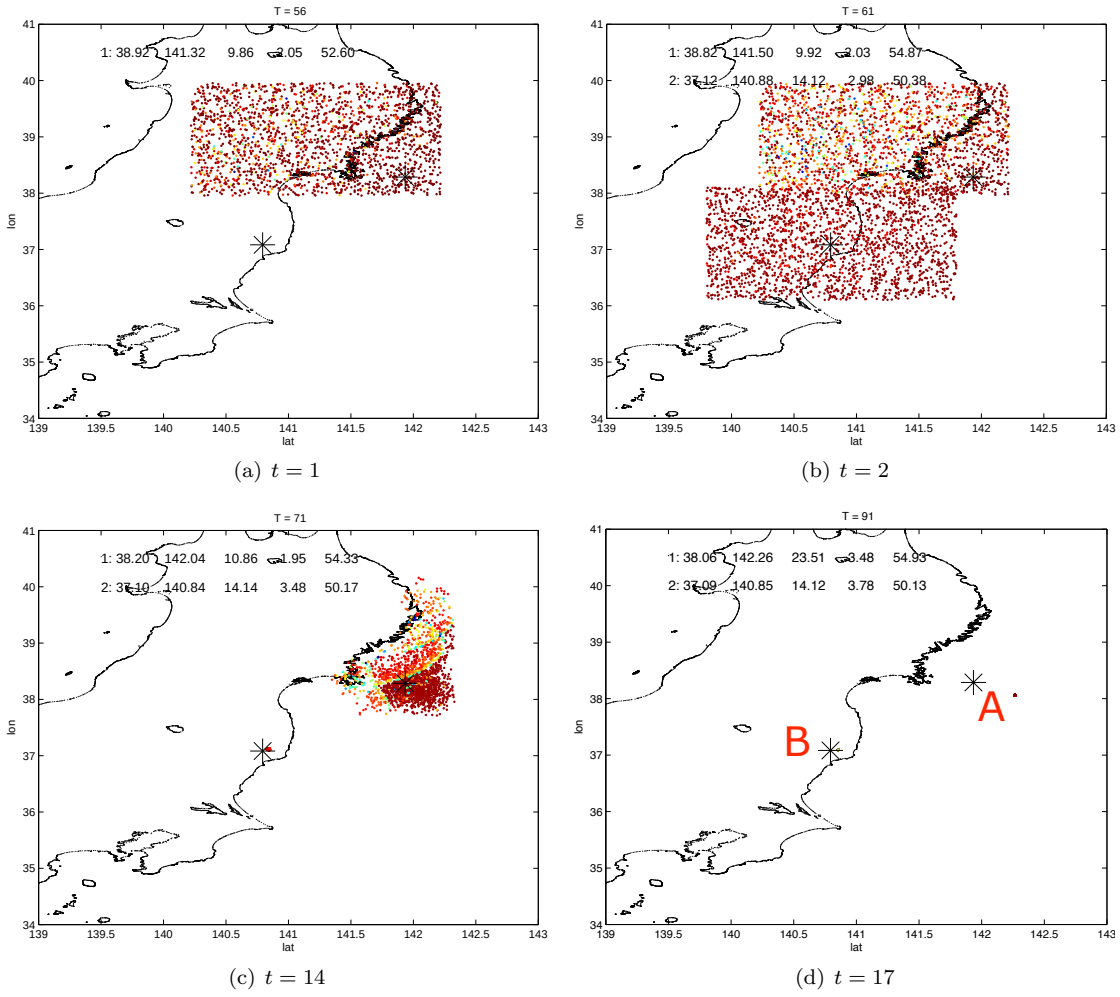


(a) EEW location estimates

提供時刻等		震源要素等					
		Seconds elapsed since first detection	震源要素				予測震度
			北緯	東経	深さ	Magnitude	
地震波検知時刻	14時19分59.3秒	—	—	—	—	—	—
1	14時20分05.9秒	6.6	38.2	142.1	30km	7.6	※1
2	14時20分10.2秒	10.9	38.2	142.1	30km	7.6	※2
3	14時20分13.9秒	14.6	38.2	142.2	10km	7.8	※3
4	14時20分20.1秒	20.8	38.2	142.2	10km	6.9	※4
5	14時20分27.5秒	28.2	37.1	140.8	10km	4.6	最大震度 3 程度以上と推定
6	14時20分29.4秒	30.1	37.1	140.8	10km	4.7	※5
7	14時20分49.1秒	49.8	37.1	140.8	10km	4.5	最大震度 3 程度以上と推定
8	14時21分09.1秒	69.8	37.1	140.8	10km	4.5	最大震度 3 程度以上と推定
9	14時21分09.7秒	70.4	37.1	140.8	10km	4.5	最大震度 3 程度以上と推定

(b) Real-time EEW records

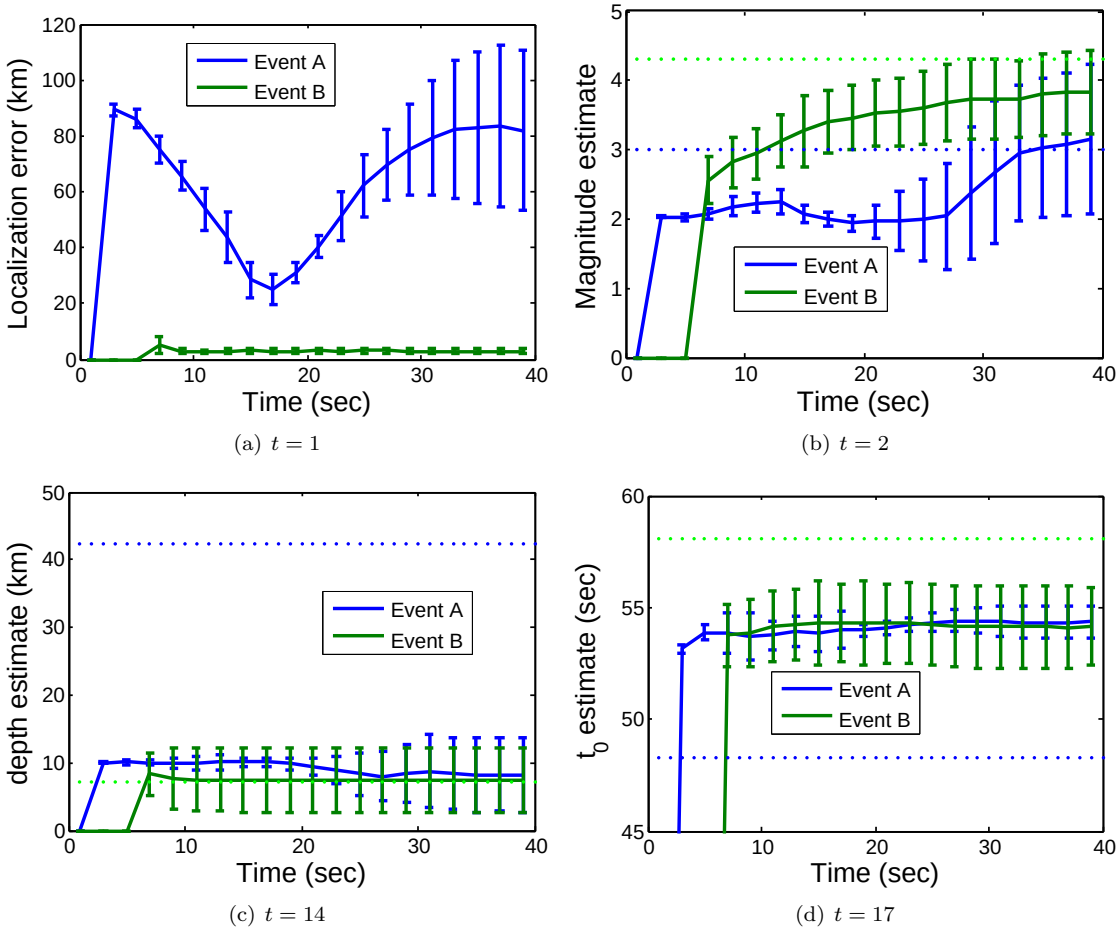
Figure 5.11: Real time warnings issued by JMA at 14:19:55, on March 20, 2011. The magnitude was over estimated by and order of 3, most likely because the two small events that happened 5 seconds apart were associated together. [Source: `http://www.seisvol.kishou.go.jp/eq/EEW/kaisetsu/joho/20110320141959/content/content_out.html`]



(a) $t = 1$ (b) $t = 2$

(c) $t = 14$ (d) $t = 17$

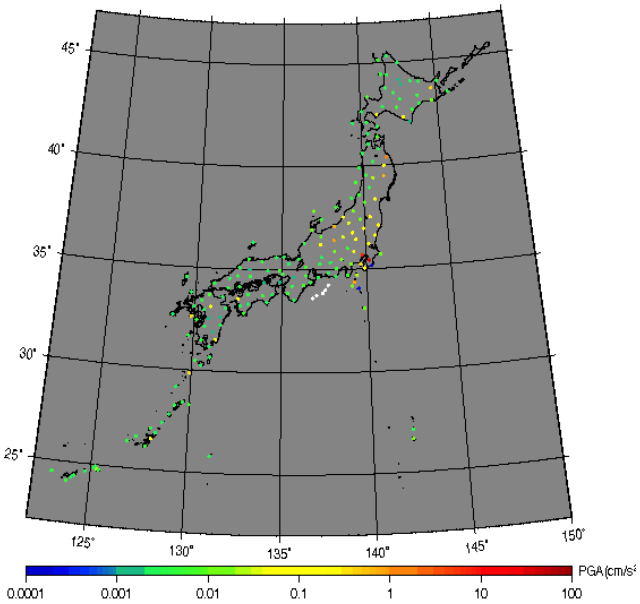
Figure 5.12: 2,000 Particles visualized in the period between 14:19:55 - 14:20:25 on March 20th, 2011. The two events are labeled in figure (d) for references. Event B converges within a couple seconds, whereas Event A takes much longer, most likely because it was much weaker and offshore.



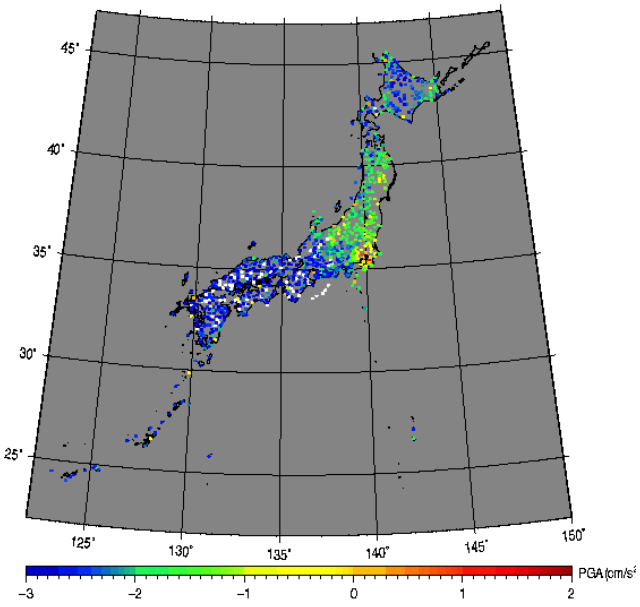
(a) $t = 1$ (b) $t = 2$

(c) $t = 14$ (d) $t = 17$

Figure 5.13: Results compiled from 20 independent runs in the period between 14:19:55 - 14:20:25 on March 15th, 2011. Post event estimates provided by JMA are marked in dotted lines. Location estimates for event B agree well with the official catalog, but this is not quite the case for event A, possibly because event A was weaker and occurred offshore.



(a) JMA stations only



(b) JMA + NIED stations

Figure 5.14: Snapshot of accelerations measured by (a) JMA stations only and (b) JMA+NIED stations at 1:36:26 on March 11 2011. There are ∼200 JMA stations and ∼800 NIED stations in Japan. Clearly, the combined network yields much richer information and better resolution for the observation of an event.

Chapter 6

Scalable Detection in Large Noisy Sensor Networks

The use of sensors in personal consumer devices has reduced the price of sensors to the point where networks of thousands and even millions of sensors are feasible. Dense networks of crowd sourcing sensors enable geospatial phenomena, such as earthquakes, to be monitored at a finer resolution than that by traditional sparse networks of high quality sensors [33, 23, 22]. The drawback is that these inexpensive sensors, limited by their costs, produce lower quality data. An important question to consider when designing a crowd-sourced networked sensing system is "can quantity compensate for quality?" In this chapter, we address this questions by exploring different data aggregation techniques for improving detection performance in large noisy networks. We start by giving examples of large-scale community-based participatory networks.

Participatory sensing. Examples of participatory sensor networks include accelerometers and GPS devices in taxi cabs for pothole detection [32] and road conditions [90], and for inferring collective daily life patterns in big cities [36]. Another example is the *iShake* project that employs smart phones as sensors to monitor the response in the event of an earthquake [2]. *The Quake-Catcher Network (QCN)* is a similar project that uses standalone MEMS devices for quick deployment after a major earthquake [23]. Most of these applications crowdsource a large amount of data through inexpensive sensors to improve environmental awareness, and to model changes that aid in scientific studies, e.g. develop a map of road conditions. In this chapter, we focus instead on a more rarely visited facet of participatory sensing; we study algorithms that *warn* about events, such as virus outbreaks [81] and earthquakes [84].

Community Seismic Network. A motivating example for this study is the *Community Seismic Network (CSN)* [1]. This project takes advantage of inexpensive standalone USB and smart phone accelerometers to build a large community sensor network in southern California. As of November

[1] `http://csn.caltech.edu`

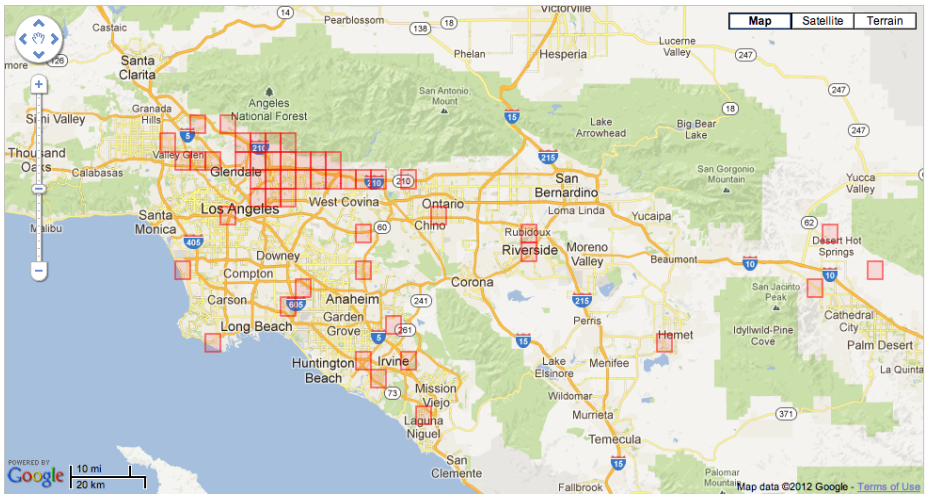


Figure 6.1: CSN clients map captured on November 26, 2012. Each box consists at least one CSN sensor. There are, on average, 250 sensors active in the whole southern California area at any given time.

2012, there are, on average, 250 stations active at any given time. Fig. 6.1 shows a real-time sensor distribution captured on November 26, 2012. Acceleration is measured at each sensor and "abnormal" activities are communicated to distributed cloud servers for processing. Since September 2011, the network has detected all 14 seismic events of M3.0 and above.

Challenges. Working with networks of massive size and scale is challenging for several reasons.

- **Electronic noise**: Sensors that can be widely distributed tend to trade precision for price. For example, the GPS location provided by a smart phone will not be as accurate as that provided by most standalone GPS devices. Sensors of lower quality have higher inherent electronic noise that yields lower quality data.

- **Ambient noise**: Such large numbers of geo-spatially dispersed sensors cannot be handled by a few professionals; instead community participation must be called into action. If the sensors are managed by untrained volunteers, depending on the installation method, the level of noise to which the sensor is exposed may vary drastically. For example, an accelerometer installed on an unstable table top in a household with small children will have very different noise profile than a professional seismometer sitting in a vault 10 feet underground.

- **Data processing**: The network infrastructure and algorithms need to handle continuous data processing from a massive number of heterogeneous sensors. In addition, community sensor networks may experience large fluctuations in sensor number and settings at any given time. For example, sensors inside a smart phone will report a different location if the user is moving. It may also be taken off the network any time when the phone is turned off.

Contributions and chapter organization. The system infrastructure and the algorithms running on it must be robust to all three challenges. In the rest of this chapter, we address these issues with a focus on *detection*, using the example of an existing community sensor network — CSN. We begin with the introduction of a scalable sensing model that will be used throughout the chapter (Sec. 6.1). We then describe, in the order of increasing complexity, intuitive aggregation algorithms for processing geospatial time-series data with the goal of improving detection performance (Sec. 6.2). Lastly, we show that by establishing the link between improved detection and existing sparsity-inducing optimization routines, we can construct a linear transformation for detection that exploits sensor correlation. This study leads to learning algorithms that outperforms several state-of-the-art algorithms (Sec. 6.4).

6.1 Sensing Model

Depending on the system architecture design, there are, in general, two types of sensing models — *streaming* and *compressed*. The two models were briefly introduced in Sec. 1.3.1. Here we give a formal definition.

6.1.1 Streaming — Centralized Detection Model

Each sensor i monitors the environment continuously and reports its observation $\mathbf{y}_i$ to a fusion center.

$$\mathbf{y}_i = \mathbf{x}_i + \epsilon_i$$

$\mathbf{x}_i$ is the uncorrupted event signal observed by sensor i. Let E be a random variable that takes the value 1 if there is an event and 0 otherwise; we assume that

$$x_i \in \begin{cases} \{\mathbb{R}^+, \mathbb{R}^-\} & E = 1 \\ \{0\} & E = 0 \end{cases}$$

The electronic noise and ambient noise of a sensor i are modeled as a single noise term $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$. This is a simplification, as ϵ_i is time and environment dependent. Without loss of generality, we can further assume that all sensors experience the same amount of noise $\epsilon_i = \epsilon, \forall i = 1, \ldots, p$

6.1.2 Compressed — Distributed Detection Model

In a massive sensor network, it is infeasible for two reasons for each sensor to constantly stream raw measurements to the fusion center. First of all, data streaming draws a significant amount of power. If the sensor is contained in a mobile device such as a smart phone, then it is unreasonable to leave

the volunteer with a drained battery within a few hours. Second, and more importantly, processing continuous raw measurements from a massive number of sensors puts tremendous demands on the fusion center and its network capacity. With these constraints, it is advantageous to distribute the computation from the fusion center to each sensor such that the sensors only communicate with the fusion center when something abnormal occurs (an *anomaly*). The process for determining abnormality is usually referred to as *anomaly detection* or *picking*, and has been studied in previous work such as [68] and [34]. We model the resulting data stream as the following under this operation mode:

Let x_i be the uncorrupted binary variable that indicates whether a sensor i detects during an event between $t-1$ and t.

$$x_i = \begin{cases} 1, & E = 1 \\ 0, & E = 0 \end{cases}$$

At time t, sensor i does local computation to determine the value of a binary random variable y_i that indicates whether something abnormal has happened.

$$y_i = \begin{cases} x_i & \text{with probability } 1-\pi \\ 1-x_i & \text{with probability } \pi, \end{cases}$$

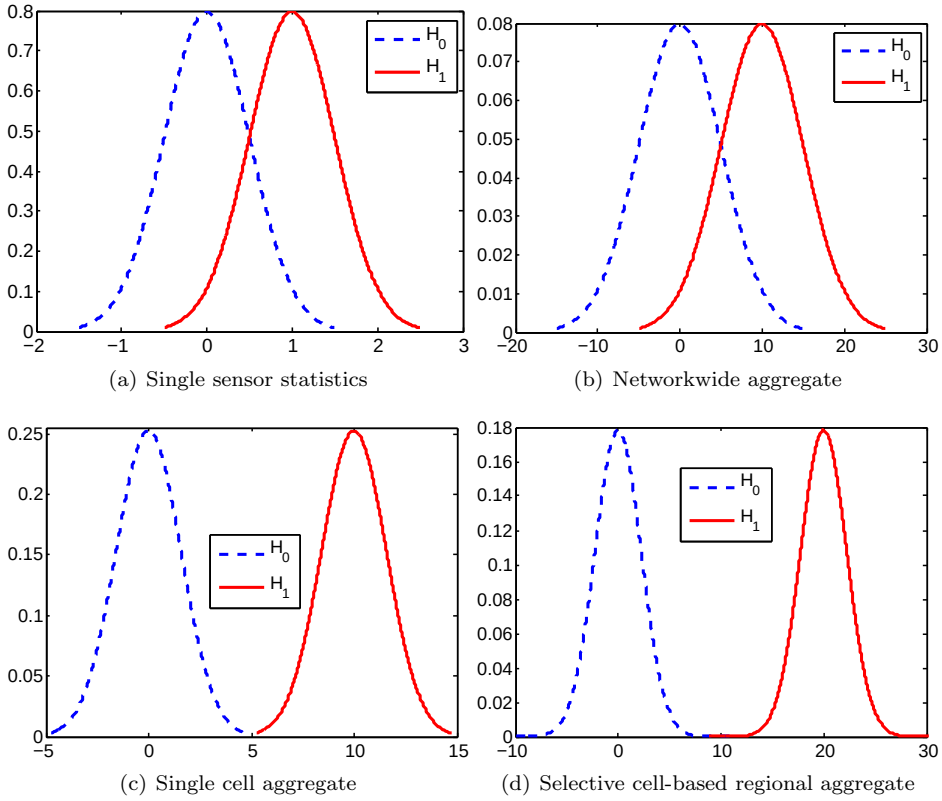
Here the electronic and ambient noise are now modeled as a binary channel noise π, i.e. a random bit flip incurred with probability π. Systems in which sensor errors are asymmetric, e.g., where the probability of a false positive is higher than that of a false negative, may give better results. The theory developed here can be extended for that case in a straightforward way.

In this model, sensors only communicate to the fusion center when $y_i = 1$. At time t, the fusion center observes a vector $\mathbf{y}^t = [y_1^t, y_2^t, \ldots, y_p^t]^T \in \{0,1\}^p$, $\mathbf{y}_i^t = 1$ if a message from sensor i is received between time $t-1$ and t. We will use this model for the rest of the chapter.

6.2 Cell-Based Data Aggregation

Detection of complex geospatial events is difficult with community sensors. Aside from the challenges mentioned in the beginning of this chapter, the event signal can be so weak that both single sensor statistics (i.e. $\max_i \mathbf{y}_i > \tau$) and naive network aggregation (i.e. $\sum_i \mathbf{y}_i > \tau$) fail to separate it from background noise. The signal-to-noise (SNR) ratio can be improved by grouping sensors that are spatially close and performing detection test on the aggregated data within each group. The amount of improvement in detection performance can be visualized with a simple example.

Suppose there are 100 sensors distributed uniformly in 10 cells. We assume that all sensors in



(a) Single sensor statistics (b) Networkwide aggregate

(c) Single cell aggregate (d) Selective cell-based regional aggregate

Figure 6.2: Illustration of simple data aggregation to improve detection performance. Both single cell and cell-based regional aggregation algorithm improve detection performance over single sensor statistics and networkwide total aggregation when observing a weak event that is only observable in part of the network.

the same cell observe the same background noise and the same event signal, i.e. the sensor readings can be treated as independently identically distributed random variable. Now suppose a weak event that is only observable in 2 out of 10 cells. Fig. 6.2 shows that while the null hypothesis $\mathcal{H}_0$ and the alternative $\mathcal{H}_1$ are inseparable using single sensor statistics (Fig. 6.2(a)) and the network-wise naive aggregation (Fig. 6.2(b)), it is more separable with the single cell-based data aggregation (Fig. 6.2(c)). A very simple way to implement the cell-based aggregation method is to partition the space into non-overlapping cells and identify for each sensor which cell it belongs to. This can be done efficiently with libraries such as the *geocell library* as described in [70].

6.3 Exploiting Spatial and Temporal Correlations

The single cell-based aggregation algorithm may not be the optimal detection strategy. For example, in the application of seismic detection, depending on the hypocenter location relative to the network, the seismic wave may affect more than one cell at a time. In such a scenario, the SNR can be further improved by aggregating data from the right groups of cells that tend to experience the first shakings

together ("co-activated"). Using the same example as in Sec. 6.2, Fig. 6.2(d) shows that combining data from two cells further increase the separability between $\mathcal{H}_0$ and $\mathcal{H}_1$. The choice of cell groupings may require human insights on the spatial and temporal relationship of the phenomenon, or can be learned from data, as we will discuss in the next section. Here, let us assume that the phenomenon is well characterized and we will show how detection performance is improved by exploiting this knowledge.

6.3.1 Cell-Based Regional Data Aggregation

A geospatial event can be very coarsely modeled as an isotropic point source event that travels in all directions at a constant speed. With this assumption in mind, we can break down the detection problem into a few case scenarios in terms of how the manifestation of the event comes into contact with the identified clusters of sensors. By exploiting sensor co-activation patterns in these scenarios, one can design a more efficient on-line event detection algorithm.

Continuing with the earthquake detection example, Fig. 6.3 shows three such possible cases after we pre-gridded the area into cells — (a) the epicenter is inside the cluster, (b) the epicenter is diagonally away from the cluster, and (c) the epicenter is on the side and away from the cluster. In each of these cases, regions of different sizes and shapes will be activated in different sequences during an event. While it is computationally nontrivial to partition a 2-dimensional space into arbitrary regions, there exists libraries that compute these geospatial regions efficiently [70].

We denote a collection of sensors in each cell-based region as c. A sensor i is in a region (or cluster) c if $i \in c$. Note that each sensor can belong to more than one region. The test computes the ratio of likelihood for the two hypothesis: a) that there is an event ($E = 1$), and b) that there is no event ($E = 0$). Assuming the distributed detection model described in Sec. 6.1, we perform following test for each region c

$$\frac{Binomial(k; |c|, E = 1)}{Binomial(k; |c|, E = 0)} \geq \tau \tag{6.1}$$

where $k = \|\mathbf{y}(\mathbb{1}_c)\|_0$, $\mathbb{1}_c$ is the support for cluster c. τ is the decision threshold chosen to satisfy a constraint on the system false positive rate. If the inequality holds for any cluster c, the system declares detection.

In a simulation setting, we compute the system-wide detection performance for each of the scenarios illustrated in Fig. 6.3 (shaded in different colors) with either 20 Phidgets (16-bit accelerometer) or 2000 Androids (12-bit accelerometer), distributed according to the population density in the area. The regions, in terms of activation sequence, are identified a priori using a geocell library [70]. A region consists of multiple nearby geocells. Each geocell is $\approx 10 \times 10\ km^2$ in size, which is approximately how far the shock wave travels in two seconds. We can then assume that all sensors in the same region have roughly the same SNR and model them as independently identically distributed

random variables. In each time step of two seconds, we perform hypothesis testing on each of the regions and compute the system-wide ROC curves. We compare this algorithm to a naive network-wide aggregation scheme, which performs the same test as in Eq. (6.1) with c being the region that includes all geocells.

6.3.2 Simulation Experiments

We collect 1000 sets of simulated measurements from 2000 Androids (lower quality sensors) and 20 Phidgets (higher quality sensors) separately during a M5.5 event within 60 km of downtown Los Angeles. During a period of $T = 0 - 10$ seconds after the event occurs, we perform the cell-based regional aggregation algorithm on each two-second interval. The results are compared to the naive networkwide aggregation algorithm discussed above.

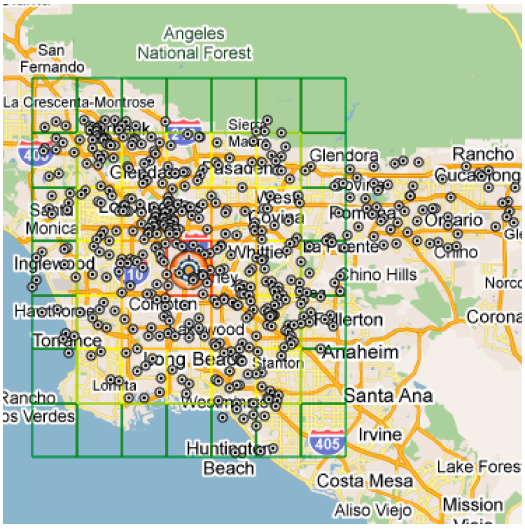
Fig. 6.4(a) shows an example of how ROC curves of 2000 Androids evolve in time for the corner case illustrated in Fig. 6.3(b). We slice the surface of this figure at the false alarm rate of 1 per year and retrieve the detection rate as a function of time. The results in Fig. 6.4(b) and Fig. 6.4(c) show that cell-based regional aggregation reaches 0.99 true positive rate on average six seconds faster than the naive network aggregation for both Android and Phidget sensors.

The improvement in detection time clearly highlights the benefit of selective regional data aggregation. It also showcases the tradeoffs between delayed decision making and gain in detection confidence. In the case with 2,000 Androids (Fig. 6.4(b)), we can initiate alarm at T=2 seconds that allows us to give tens of seconds of early warning to surrounding cities, such as Santa Barbara, CA or San Diego, CA, but with only 20% confidence. Or we can wait till T=10 second or after to issue the alarm with $\sim 100\%$ confidence.

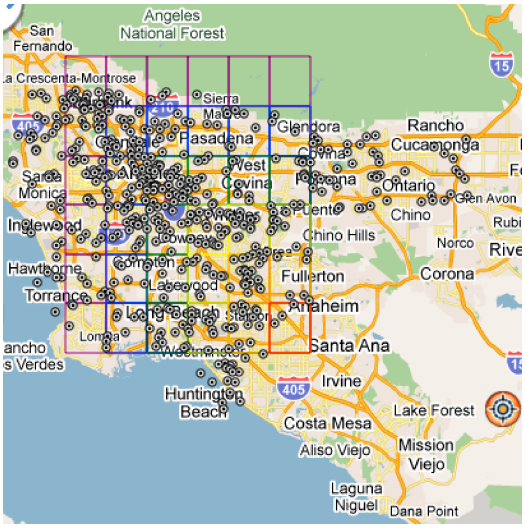
With the strong assumptions on simple geometry about a highly complex geospatial event such as earthquake, this study serves to provide insights on how intelligent grouping of correlated sensors can improve detection performance. We will further explore in the next section how these correlations can be constructed automatically without domain expertise about the phenomenon.

6.4 Learning Sensor Correlations for Detection

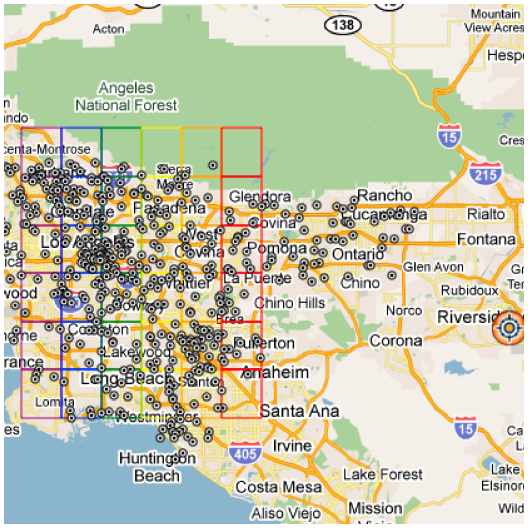
In Sec. 6.2 and Sec. 6.3, we considered different methods for reliable detection of geospatial events in large community sensor networks and showed that, while the signal to noise ratio may be small in the network as a whole, by organizing the correlated sensors into groups and comparing the linear combination of sensor data in each group, we can improve detection performance. Next, we formalize the idea that the signals can be concentrated via grouping, and present a class of algorithms that learn from historic or simulated data how to form such groups to guarantee improved detection.



(a) Central Model

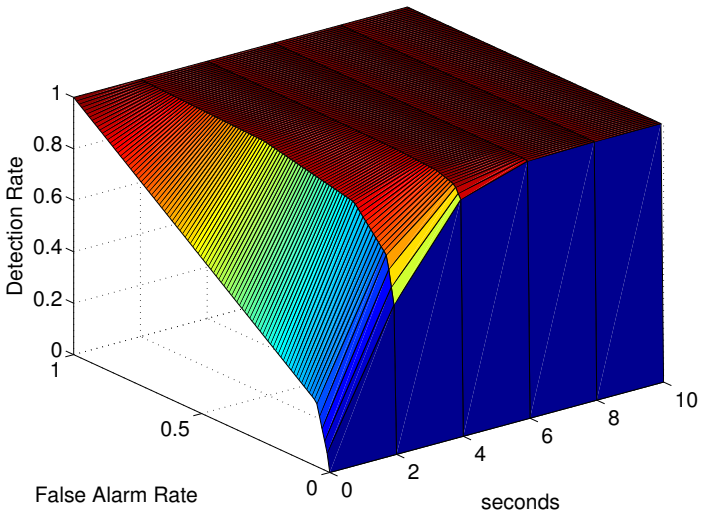


(b) Corner Model

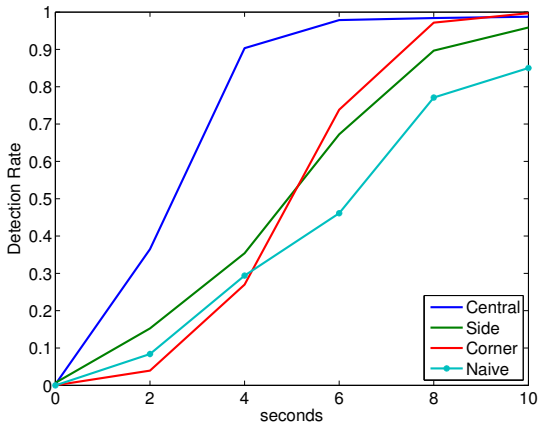


(c) Side Model

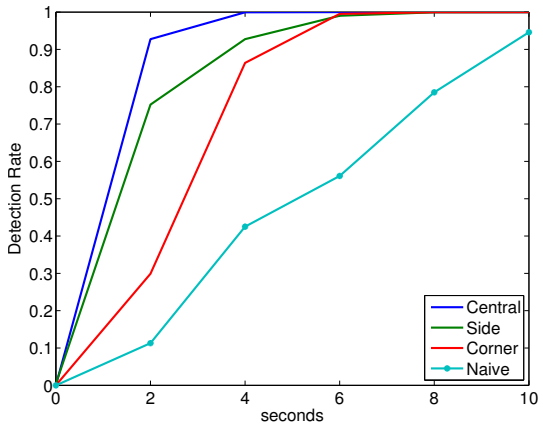
Figure 6.3: Regions of different sizes and shapes are activated in different sequences for each of the three scenarios. The rainbow-colored rings indicate the order of activation. Red: first. Purple: last. Hypothesis testing on aggregated measurements within each region improves detection performance in comparison to naive network aggregation.



(a) Surface plot of ROC curves as a function of observation time for the corner scenario with 2000 Androids.



(b) 2000 Androids



(c) 20 Phidgets

Figure 6.4: Detection of a M5.5 event with (b) 2,000 Androids and (c) 20 Phidgets in the three scenarios illustrated in Fig. 6.3. The detection rate (true positive rate) is computed to guarantee at most one false alarm per year at the system-wide level. Results computed using the geocell-based association algorithm are compared to those using the naive network-wide aggregation algorithm. Setting a desired detection rate at 0.99, it is clear that by exploiting the prior knowledge on sensor correlations, we can, on average, detect 6 seconds faster.

6.4.1 Sparsifying Basis

As a review, a basis can be viewed as a transformation that maps the original signal into a different coordinate system by multiplying the signal matrix with a basis matrix. Intuitively speaking, if a transformed signal is "simpler" in this new coordinate system (i.e. has fewer non-zeros terms which means the original signal can be represented with fewer components than its original dimension), then we say the signal is *sparsifiable* and the basis is *sparsifying*. More concretely, if $\mathbf{x}$ is the signal vector, and $\mathbf{B}$ is a sparsifying basis, we have $||\mathbf{B}^T\mathbf{x}||_0 \ll ||\mathbf{x}||_0$.

One way to construct a sparisifying basis is through hierarchical clustering [82, 53]; when two smaller clusters c_l, c_r are merged into a new cluster in the clustering process, a basis element $\mathbf{b}_i$ is created by taking the unit vector defined over c_l and c_r

$$\mathbf{b}_i \propto \left(\frac{1}{|c_l|}\mathbb{1}_{c_l} - \frac{1}{|c_r|}\mathbb{1}_{c_r} \right) \tag{6.2}$$

where $\mathbb{1}_c$ is the support for cluster c. $p-1$ basis elements are created in this way with $p-1$ merges during the hierarchical clustering process. The p^{th} basis element is defined by the equivalence of a global average.

$$\mathbf{b}_p = \frac{1}{\sqrt{p}}\mathbb{1}_p$$

All the column basis elements $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_p$ form an orthonormal matrix. This `haar` wavelet-like basis is shown to be sparsifying for event signals generated according to a latent tree model [82]. It was further shown that the sparsification leads to improved detection under weak signal and strong noise for both tree-structured [82] and graph-structured activation patterns [53] such that the null hypothesis $\mathcal{H}_0$ is better separable from the alternative $\mathcal{H}_1$ with high probability as the number of sensor p goes to infinity.

The link between sparsity and improved detection was further strengthened in [35] when Faulkner *et al.* introduced the "sparsification ratio" $\frac{||\mathbf{x}||_0}{||\mathbf{B}^T\mathbf{x}||_0}$ and showed how it affects asymptotic detection performance. The next section develops upon this idea to explore other possible sparsifying bases.

6.4.2 Learning Sparsifying Basis

Results in [35] show that if an event is "sparsifiable", then it is possible to construct a linear transformation matrix $\mathbf{B}$ that projects the original measurements onto a simpler ("sparser") representation in a different coordinate system that better separates $\mathcal{H}_0$ and $\mathcal{H}_1$, even in the presence of strong noise. The `haar` wavelet is an example of such a sparsifying basis, assuming that the data has certain structured dependencies. For other types of data, can we still construct similar sparsifying bases? More importantly, can we learn such bases from data without imposing assumptions on data dependencies?

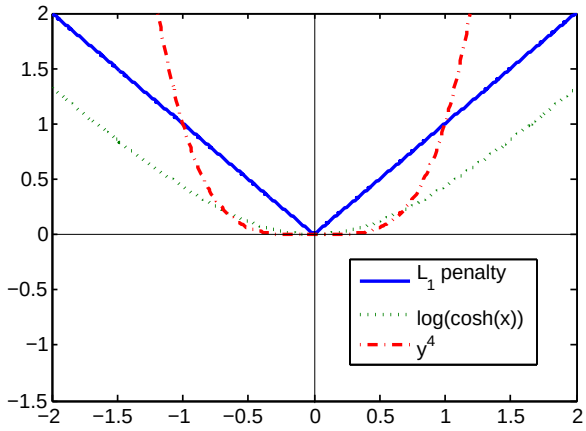


Figure 6.5: Smooth ℓ_1 approximation functions used in ICA with the linear ℓ_1 penalty function plotted in blue solid line.

Let $\mathbf{B}$ be an orthonormal matrix and $\mathbf{x}$ a vector of uncorrupted binary activations, Theorem 1 in [34] states that the sparsification ratio $\frac{||\mathbf{x}||_0}{||\mathbf{B}^T\mathbf{x}||_0}$ directly impacts the amount of separation between $\mathcal{H}_0$ and $\mathcal{H}_1$. In fact, given that $||\mathbf{x}||_0$ is fixed, the two hypotheses are maximally separated when the transformed $||\mathbf{B}^T\mathbf{x}||_0$ is minimized. In other words, we can construct the optimal basis by solving the following optimization problem:

$$\arg\min_{\mathbf{B}} ||\mathbf{B}^T\mathbf{X}||_0, \text{ subject to } \mathbf{B}\mathbf{B}^T = \mathbf{I} \tag{6.3}$$

where $\mathbf{X}$ is a matrix that contains binary observations $\mathbf{x}$ as its columns and $||\cdot||_0$ is the sum of non-zero elements in the matrix.

However, direct minimization of $||\mathbf{B}^T\mathbf{X}||_0$ is NP-hard in general [27] and the ℓ_0-norm is often relaxed and replaced with the continuous ℓ_1-norm [17], which is also a common sparsity measure. The optimization task in Eq. (6.3) can then be approximated as

$$\arg\min_{\mathbf{B}} ||\mathbf{B}^T\mathbf{X}||_1, \text{ subject to } \mathbf{B}\mathbf{B}^T = \mathbf{I} \tag{6.4}$$

where $||\cdot||_1$ is the maximum absolute column sum of the matrix.

Direct approximation. Eq. (6.4) can be directly approximated by *Independent Component Analysis* (`ICA`), which solves the following optimization problem

$$\arg\min_{\mathbf{B}} G(\mathbf{B}^T\mathbf{X}), \quad \text{subject to} \quad BB^T = \mathbf{I} \tag{6.5}$$

where G is a nonlinear convex smooth approximation to the ℓ_1 penalty function, e.g. $\log\cosh(x)$, $-\exp(-x^2/2)$, and x^4 [46]. Fig. 6.5 illustrates these functions in relation to the linear penalty function.

Eq. (6.5) can be solved with stochastic gradient algorithm by taking the derivative of G. However this approach is often slow and requires fine tuning; this leads to the development of "FastICA", an efficient fixed-point algorithm. Implementation details of FastICA and in-depth analysis can be found in [46]. Let $g = G'$, the one unit algorithm for FastICA is given in Alg. 2 for completeness.

Algorithm 2: `ICA` one-unit solution

$\mathbf{b} \leftarrow$ random unit vector
while $\mathbf{b}$ *not converged* **do**
  $\mathbf{b} \leftarrow \mathbb{E}\left[\mathbf{x}g(\mathbf{b}^T\mathbf{x})\right] - \mathbb{E}\left[g'(\mathbf{b}^T\mathbf{x})\right]\mathbf{x}$;
  $\mathbf{b} \leftarrow \mathbf{b}/\left\|\mathbf{b}\right\|$;

There exists two variations of FastICA, both of which use Alg. 2 as subroutine. The *deflationary orthogonalization* greedily finds and fixes one component $\mathbf{b}_i$ at a time. The *symmetric orthogonalization* finds all components $\mathbf{b}_i, \forall i = 1, \ldots, p$ and orthogonalizes all of them at the end of each iteration [45]. While the *symmetric* approach utilizes parallelization and is multitudes faster than *deflationary* approach, the latter gives much better results empirically.

Noise-tolerant relaxed approximation. Ideally, we want to learn from noise-free observations $\mathbf{X}$. However, training data constructed from real-world measurements often contain noise or outliers, and, instead, we are forced to train with $\mathbf{Y}$, which is the observation matrix $\mathbf{X}$ corrupted with noise. In such cases, optimizing $\mathbf{B}^T\mathbf{Y}$ as in `ICA` may lead to the discovery of false correlations. Alternatively, we introduce a latent sparse matrix $\mathbf{Z}$ and relax the problem in Eq. (6.4) to the following formulation:

$$\arg\min_{\mathbf{B},\mathbf{Z}} \|\mathbf{Y} - \mathbf{B}\mathbf{Z}\|_F^2 - \lambda\|\mathbf{Z}\|_1, \text{ subject to } \mathbf{B}\mathbf{B}^T = \mathbf{I} \tag{6.6}$$

where $||\cdot||_F$ is the Frobenius norm or the squared root of the sum of element-wise squares.

$\mathbf{Z}$ can be viewed as the underlying sparse data representation in the transform domain. In other words $\mathbf{Z} = \mathbf{B}^T\mathbf{X}$. With this insight, Eq. (6.6) essentially balances the difference between $\mathbf{Y}$ and $\mathbf{X}$ and the sparsity of $\mathbf{Z}$ via tuning of the sparsity penalty constant λ, providing resilience to noise.

Although Eq. (6.6) is non-convex, fixing either $\mathbf{B}$ or $\mathbf{Z}$ makes the objective function with respect to the other convex. The objective can then be solved in an iterative two-step convex optimization process — *Orthogonal Procrustes* [38] and *LASSO with orthonormal design* [12]. The two-step procedure is given in Alg. 3 for reference. The formulation of Eq. (6.6) and the solution in Alg. 3 is equivalent to *Sparse Latent Semantic Analysis* (`SLSA`) [19], which was introduced for applications involving topic models. Here we adopt the name for consistency.

6.4.3 Implementation Issues

In this section, we describe practical issues necessary for using a sparsifying basis for event detection in real-world sensor networks.

Algorithm 3: SLSA two-step convex optimization procedure

Step 1: Orthogonal Procrustes
Fix $\mathbf{Z}$, solve $\min_{\mathbf{B}} ||\mathbf{Y} - \mathbf{BZ}||_F^2, : \mathbf{BB}^T = \mathbf{I}$
$M \leftarrow \mathbf{YZ}^T$;
$M = U\Sigma V^T$;
$\mathbf{B} \leftarrow UV$;

Step 2: LASSO with orthonormal design
Fix $\mathbf{B}$, solve $\min_{\mathbf{Z}} \|\mathbf{Y} - \mathbf{BZ}\|_F^2 + \lambda\|Z\|_1$
$K \leftarrow \mathbf{Z}^T\mathbf{Y}$;
$\mathbf{Z} \leftarrow \text{sign}(K) \times \max(|K| - \lambda)$;

Basis learning with insufficient training data. Learning a basis for p sensors requires at least p measurements of the network. If this is not available (e.g. a seismic network with 1,000 sensors may not yet have observed 1000 earthquakes), then simulations provide a practical way to supplement real data. One advantage of using simulations in this way is that while simulations may be slow and computationally intensive, the learned basis produces a fast and efficient detection rule. In Sec. 6.4.4, we empirically assess the amount of data required to train a good basis and present two case studies using only data generated from simulations.

Selecting the detection threshold. An event is reported whenever $|\mathbf{b}_i^T\mathbf{y}| \geq \tau$ for any non-constant $\mathbf{b}_i \in \mathbf{B}$. The threshold τ is typically chosen as a value that satisfies constraints on the false positive rate during cross validation with the historical data of event observations. This approach does not rely on positive training examples, and so a threshold τ can be learned using only the noise profile of each sensor. Suppose sensors $i = 1, 2, \ldots, p$ have binary error rates $\pi_1, \ldots, \pi_p$, we have $\mathbb{E}\left[|\mathbf{b}^T\mathbf{y}|\right] = \sum_i^p b_i\pi_i$. Given that the basis is orthonormal, under $\mathcal{H}_0$, Hoeffding's Inequality states that

$$\mathbb{P}\left[|\mathbf{b}^T\mathbf{y}| > \tau\right] \leq \exp\left(-2\left(\tau - \mathbb{E}\left[|\mathbf{b}^T\mathbf{y}|\right]\right)^2\right).$$

By setting the right hand side to a false positive rate constraint, we can easily derive a threshold that satisfies the system requirement. In particular, in order to ensure that $|\mathbf{b}^T\mathbf{y}| \leq \tau$ for all $\mathbf{b} \in \mathbf{B}$ (i.e., no false alarm happens) with probability at least $1 - \delta$, it suffices to choose

$$\tau = \max_{\mathbf{b}\in\mathbf{B}} \mathbb{E}\left[|\mathbf{b}^T\mathbf{y}|\right] + \sqrt{\frac{1}{2}\log\frac{p}{\delta}}.$$

This approach is similar in flavor to the threshold selection method in [34].

6.4.4 Experiments

We empirically evaluate the detection performance of the three sparsifying bases: `SLSA`, `ICA`, and hierarchical wavelets (`haar`) trained and tested on both simulated and real measurements in different

domains. The experimental setup is summarized here.

Baseline algorithms. We compare against

- `avg`: network-wide average, $1/p \sum_i^p y_i$;
- `max`: single sensor maximum, $\max_i y_i$;
- `SS`-k: scan statistics that aggregates the k-nearest neighbors for each sensor [66];
- `SS`-r: scan statistics that aggregates all sensors within a radius r for each sensor [66].

Evaluation data sets. The data sets include

- **Synthetic** data from latent tree model, 1296 nodes;
- **Gnutella P2P network**: 1769 nodes;
- **Japan seismic network**: 721 nodes;
- **CSN seismic network**: 128 nodes;
- **Long Beach** seismic network: 1,000 nodes.

Evaluation metrics and goals. We adopt two metrics in the evaluation of detection performance:

- AUC_f: measures the area-under-curve (AUC) in the Receiver Operating Characteristic (ROC) curve only for false positive rates between 0 and $f, f \leq 1$. The integral AUC_f takes values in $[0, f]$ and is normalized to 1 for simplicity. E.g. $AUC_{0.05} = 0.8$ shows that the detection performance reaches 80% of the optimal under the false positive constraint of five false alarms in 100 tests.

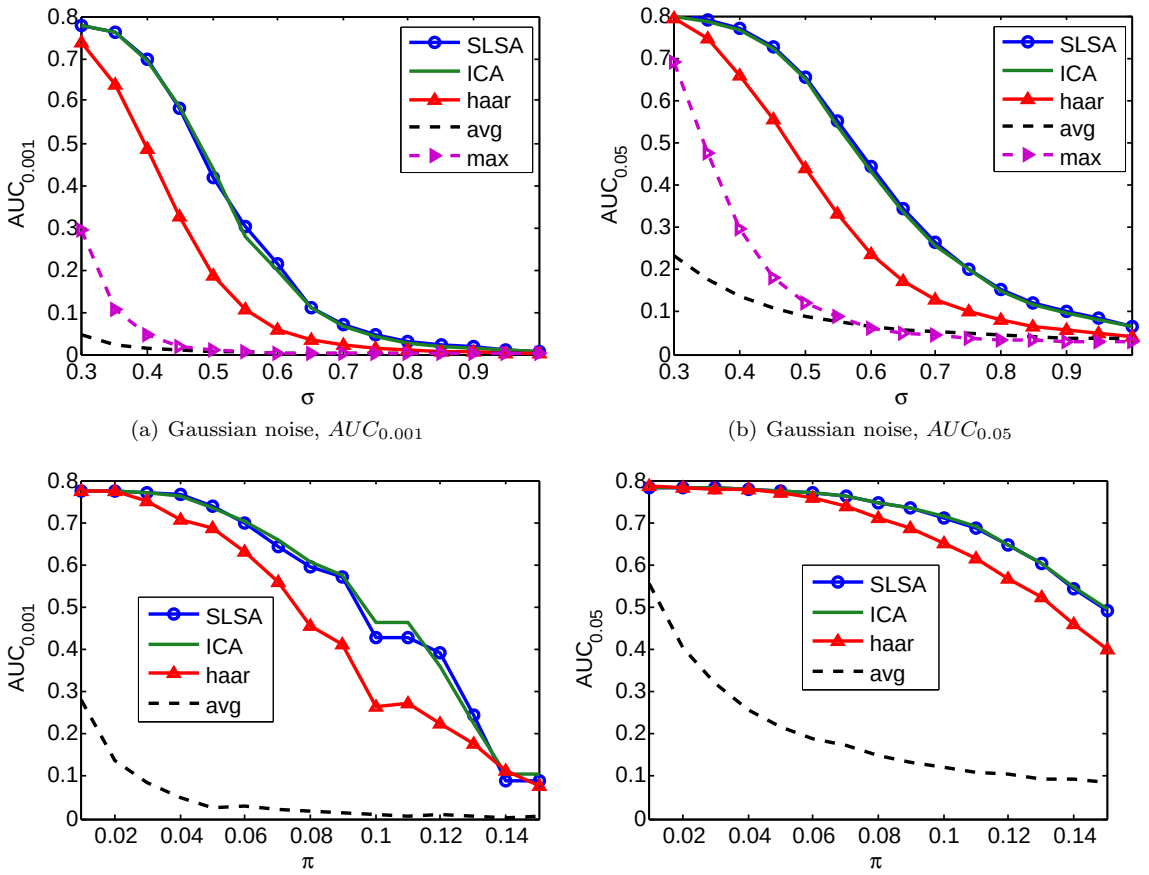
- *Detection time*: the time it takes for the test statistics to exceed a threshold that is selected to satisfy a certain system false positive requirement.

6.4.4.1 Synthetic Network

We generate samples from the latent tree model for network activation as described in [35]. The tree contains $p = 1296$ leaf nodes with degree $d = 6$ and depth $L = 4$. We choose the sparsifying parameters $\alpha = 0.5$ and $\beta = 0.95$ so that that the expected number of total activations $||\mathbf{x}||_0 < \sqrt{p}$. Of the three bases, `haar` is constructed from the known tree model whereas `ICA` and `SLSA` are trained with 20,000 samples drawn from the model. The bases are tested on 20,000 separate samples corrupted with Gaussian or binary channel noise.

Fig. 6.6 shows that all three bases outperform the naive baselines under both Gaussian and binary noise. Note that, perhaps surprisingly, both the learned `ICA` and `SLSA` outperform `haar` even though the latter is constructed from the known latent tree model.

Next we study how the network size and the number of training samples affect the quality of learned basis and detection performance.



(a) Gaussian noise, $AUC_{0.001}$

(b) Gaussian noise, $AUC_{0.05}$

(c) Binary noise, $AUC_{0.001}$

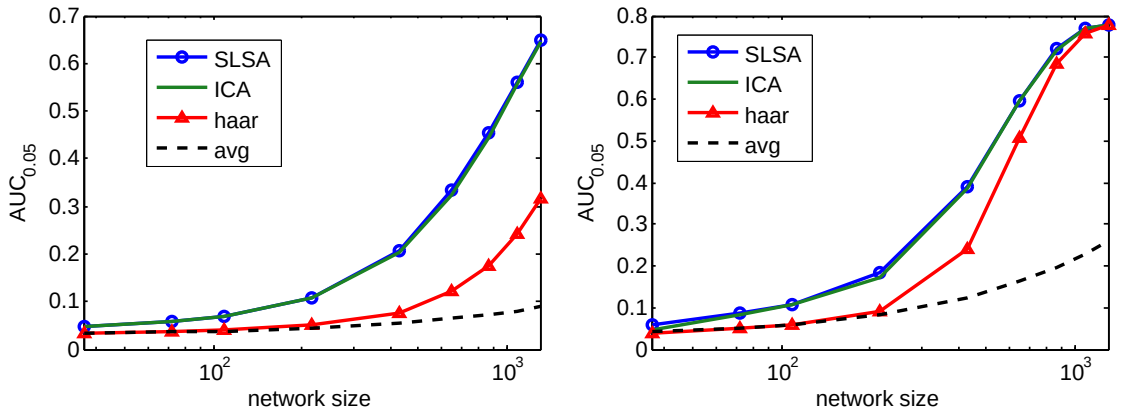
(d) Binary noise, $AUC_{0.05}$

Figure 6.6: Comparing the three bases — `SLSA`, `ICA`, `haar` to baselines — global average (and single max in (a) and (b)) on a synthetic data set generated from the latent tree model. The first and second row evaluate two different false positive constraints. The learned bases significantly outperform the baselines under strong noise.

Increasing network size. We perform basis learning with subsets of the network, using $p = [36, 72, 108, 216, 432, 648, 864, 1080, 1296]$ sensors and $n = 20000$ training samples. Fig. 6.7 shows that the detection performance of the learned bases grows more than five times faster than the baseline. Note that `haar` is now learned from data; this accounts for the slight inferior performance compared to that in Fig. 6.6.

Increasing number of training samples. With the network size fixed, we learn the bases for increasing numbers of training samples $n = [20, 100, 200, 1000, 2000, 4000, 10000, 15000]$. Fig. 6.8 shows that `haar` outperforms at smaller training size since it assumes a simple hierarchical structure. It also shows that it takes only 2,000 samples for `ICA` and `SLSA` to achieve the same detection performance as using all 20,000 samples.

Training time. Fig. 6.9 plots the time it takes to train the three bases in the above experiments. `ICA` takes roughly 5-10 times longer to train than `SLSA`.



(a) Gaussian noise, $\sigma = 0.5$ (b) Binary noise, $\pi = 0.04$

Figure 6.7: Detection performance as a function of network size $p = [36, 72, 108, 216, 432, 648, 864, 1080, 1296]$ using all 20,000 training samples. (a) and (b) show that the detection performance of the learned bases grows more than 5x faster than the baseline.

6.4.4.2 Gnutella P2P Network

Our next set of experiments is on simulated virus outbreaks on a peer-to-peer network. We obtain a snapshot of the Gnutella P2P file sharing network[2] through the Stanford Network Analysis Project (SNAP). 1769 nodes of the highest degree of connectivity were selected from this network for the experiment. Fig. 6.10 visualizes part of this sub network. We simulate 40,000 outbreak events – "*cascades*" – that mimic virus outbreaks on this directed network. We adopt the independent cascade model, where a starting node is picked at random, and whenever a node r is infected, a connected node w is infected with decreasing probability as a function of distance to r.

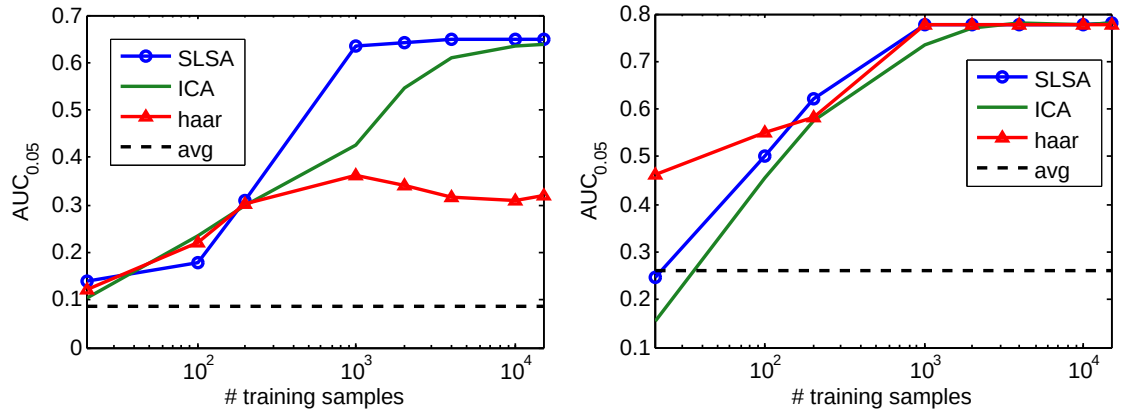
Here `haar` is constructed as a spanning tree wavelet basis, assuming known network structure [53] using Wilson's uniform spanning tree (UST) sampling method on a directed graph via random walk [86]. We also apply baseline `SS`-k is for reference. k is selected based on the prior knowledge that in the cascade model on average between 10 and 30 nodes are activated in each event.

Fig. 6.11(a) and Fig. 6.11(b) compare the detection performance evaluated on 40,000 testing samples. Both `SLSA` and `ICA` demonstrate superior detection performance compared to the state of the art algorithms that use additional prior knowledge of the network.

6.4.4.3 Japan Seismic Network

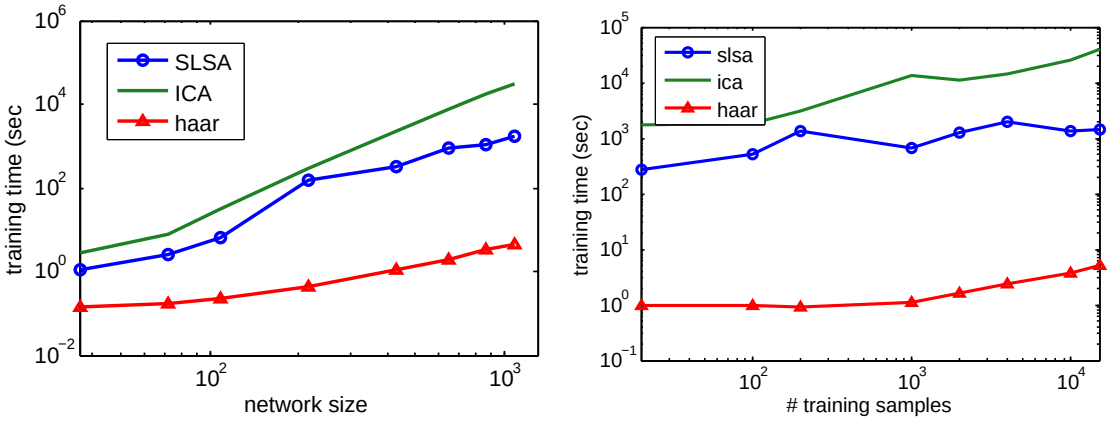
Next we turn to perhaps one of the most robust and long-running sensor networks in the world – the Japan seismic network. We obtain 48-hour, 150 GB of recordings from 721 *Hi-net* seismometers (locations shown in Fig. 6.12(a)) for the dates March 18 and 19, 2011, just one week after the Tohoku M9.0 earthquake on March 11, 2011. On both days, over 1,000 events that ranged from

[2] `http://snap.stanford.edu/data/p2p-Gnutella05.html`



(a) Gaussian noise, $\sigma = 0.5$ (b) Binary noise, $\pi = 0.04$

Figure 6.8: Detection performance as a function of training data size. (a) and (b) show that it only takes approximately 2,000 samples for both `ICA` and `SLSA` to achieve the same performance as using all 20,000 samples.



(a) (b)

Figure 6.9: Training time comparison. On average, `ICA` takes 2-10 times longer to train than `SLSA` in both experiments with network size and with training sample size.

M1.0 - M6.0 were recorded in the the Japan Meteorology Agency catalog[3]. Many events triggered clustered activations as observed in Fig. 6.12(b).

For all 1795 events recorded on March 18, 2011, 10 snapshots of network activations at a two-second period were taken after the first detection at each event to construct the training data set of [p x n] = [721 x 17950]. The learned bases are tested on the first one-second data of the 1324 events recorded on March 19, 2011. We added binary noise of different error rate to control the problem complexity.

For the comparison with the `SS`-r baseline, the aggregation distance r is selected to be 20km which is roughly the distance covered by the seismic waves in a 2-second period. Fig. 6.13 presents

[3] `http://www.hinet.bosai.go.jp/REGS/JMA/?LANG=en`

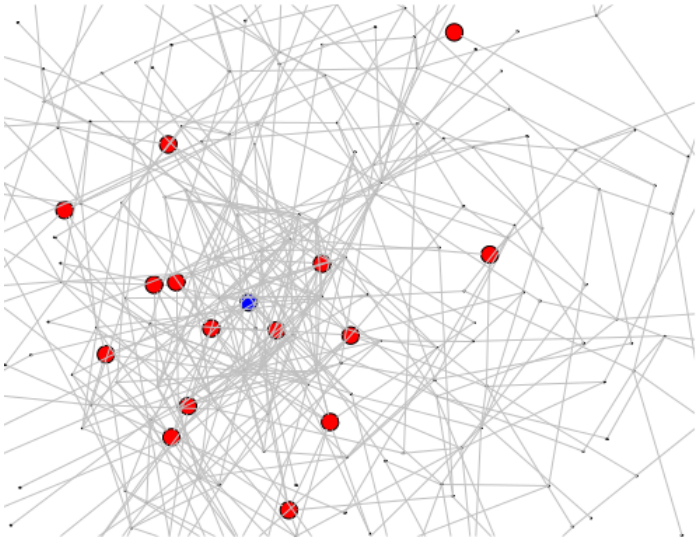
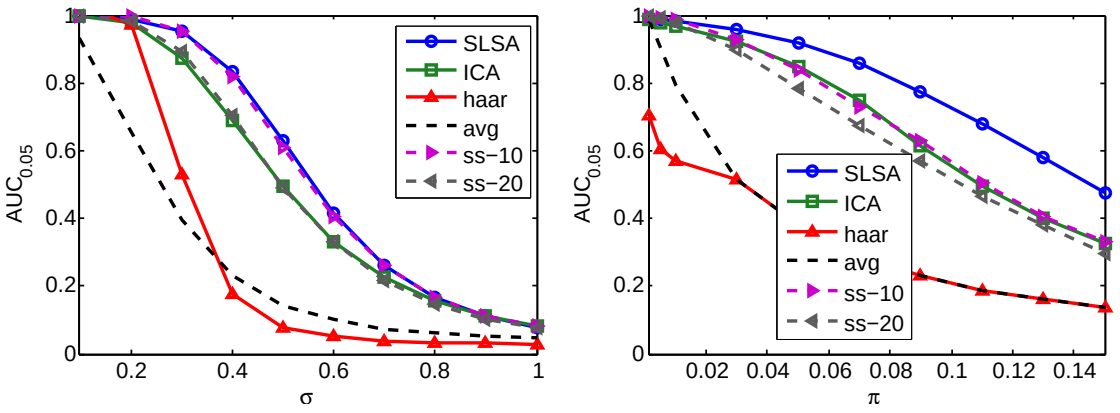


Figure 6.10: Visualization of $\sim 1/10$ the total network with a sample activation pattern colored. Blue: first infected node, Red: nodes subsequently infected through the cascade.

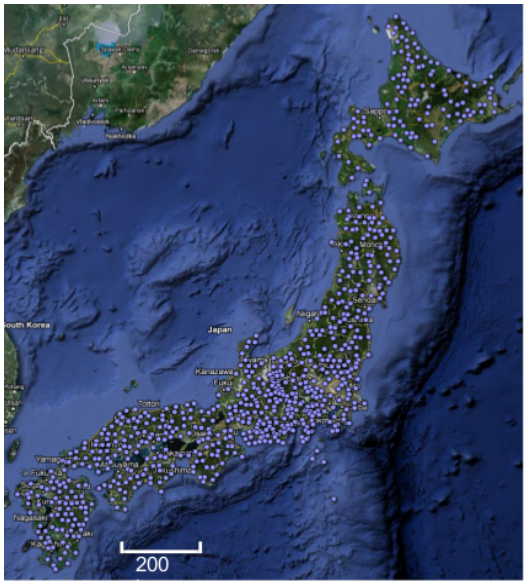


(a) Gaussian noise

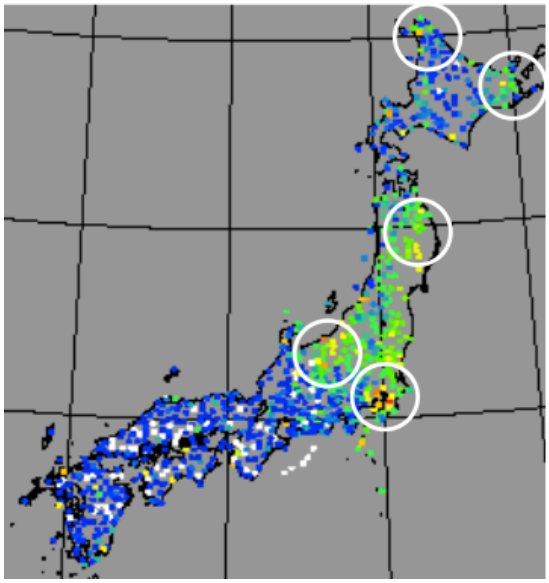
(b) Binary noise

Figure 6.11: Experiment with Gnutella P2P network. (a) and (b) show that the learned bases achieve and exceed the state of the art algorithms that use additional prior knowledge of the network.

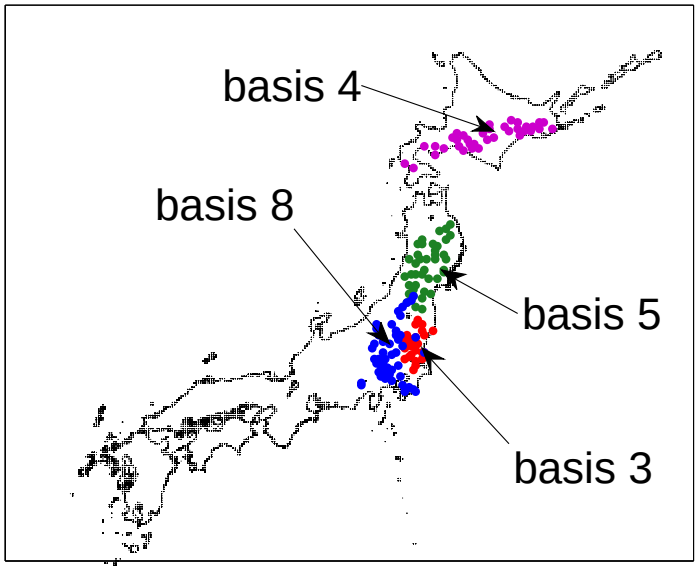
the performance in detecting within two seconds of event arrival under a very small false positive constraint of 0.001. Of the three learned bases, both `ICA` and `SLSA` show significant gain in detection power, whereas `haar` has no improvement over the `avg` baseline. Perhaps surprisingly, `SS`-r20 performs very poorly in comparison. An explanation is that most of the events during this period originated from the ocean and affected an array of stations along the coast. However, this pattern is not captured by the fixed radius subset scan construction. This explanation is supported by the plot of four prominent basis elements from `ICA` in Fig. 6.12(c). This example demonstrates the limited detection capability of subset scan for unknown patterns, and the power of learning-based detection algorithms such as `ICA` and `SLSA`.



(a) 721 Hi-net stations in Japan



(b) Clustered activations are frequent in the period of the experiment.



(c) Four prominent ICA basis

Figure 6.12: Japan's seismic network. The 721 Hi-net stations in (a) frequently exhibit localized activation patterns as circled in (b), which plots raw accelerations (red: large shaking, blue: small shaking). The learned bases are able to capture these nonlinear patterns with basis elements such as the ones in (c).

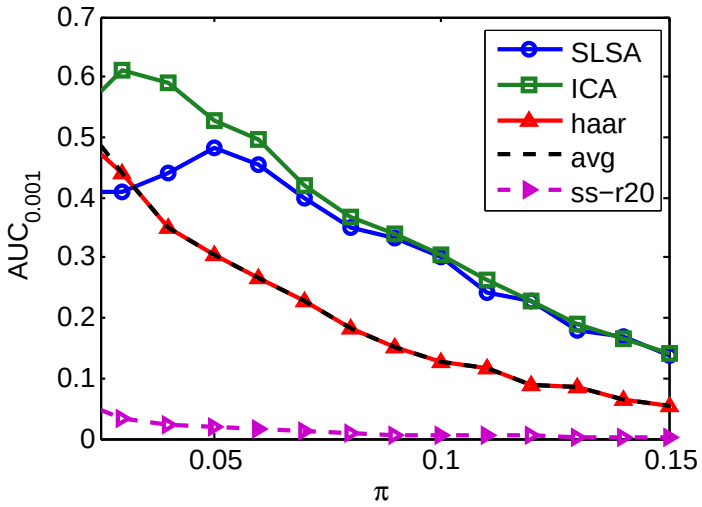


Figure 6.13: Japan data detection results. The learned bases demonstarte 2x better detection performance compared to the baselines, while algorithms with hard-coded patterns, such as SS-r20, fail to perform well.

6.4.4.4 Dense and Participatory Seismic Networks

Lastly, we return to our motivating examples of dense participatory sensor networks, using two real-world dense seismic networks in Southern California as examples. We show that bases can be learned using earthquake simulators to generate the binary activation patterns for training and select detection thresholds based on a sensor noise profile, as discussed in Sec. 6.4.3. Given the shortage of testing data – only a small number of events have been recorded by these networks, not enough to reliably compute AUC scores – the detection performance is evaluated in terms of detection time as described in the beginning of this section. This measure of time is critical to many applications; sub-second savings may prevent a huge loss of capital and lives.

Generating training data. To generate training data, we use a basic earthquake simulator. First, a prior distribution of seismic events in Southern California is constructed from a list of historic earthquakes (Fig. 6.14) compiled from the USGS database[4]. Training data is generated by randomly sampling a hypocenter from this distribution, and then simulating sensor detections. Time sequences of sensor activations are generated from a model based on the speed of seismic waves and distance to the hypocenter. This model is simplistic, but it executes quickly. An activation probability similar to that in [56] is used to simulate signal attenuation for unreliable noisy sensors.

Community Seismic Network. We simulate network activation snapshots for 128 Community Seismic Network [22] sensors based on 1000 random simulations as described above. The algorithms are evaluated with four recent events recorded by the actual network, as shown inFig. 6.15(a). Fig. 6.15(b) shows that the learned bases generally achieve faster detection than do other algorithms,

[4] http://earthquake.usgs.gov/earthquakes/eqarchives/epic/

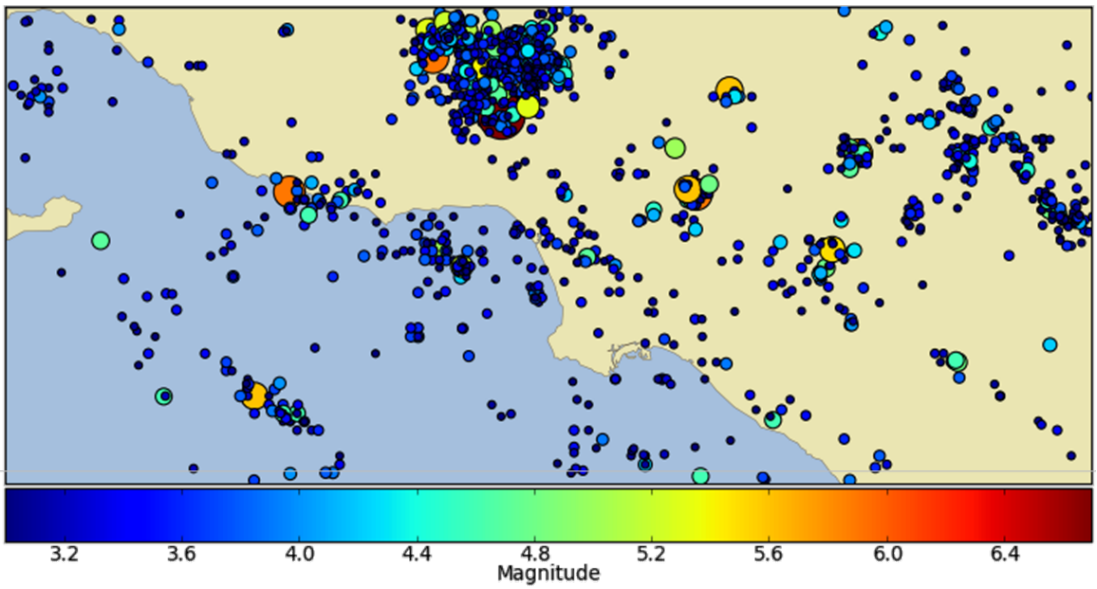


Figure 6.14: Quakes in southern California since 1973

e.g. 8 seconds faster in detecting the Beverly Hills event. Note that `ICA` performs better than `SLSA`, as simulations are noise-free.

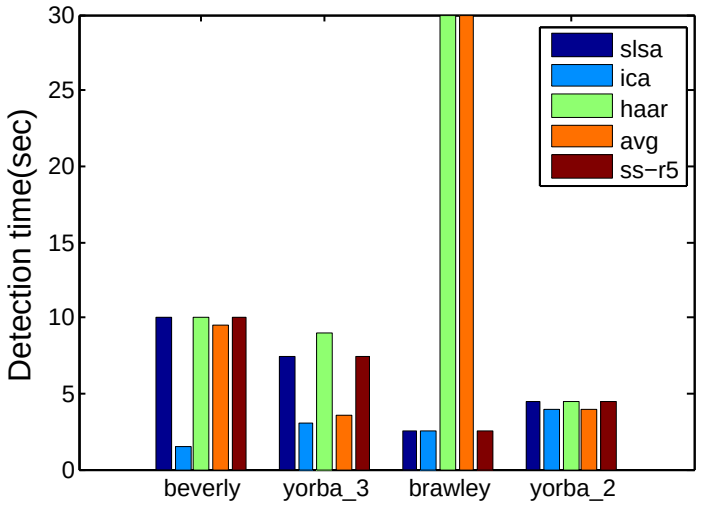
Long Beach Array. The Long Beach network consists of approximately 5000 sensors that cover an area of 5 x 7 km^2. The network was deployed for six months during the first half of 2011 to provide detailed images of the Signal Hill Oil Field in Long Beach, California. During the deployment period, a total number of five detectable earthquakes were recorded by the network, as shown in Fig. 6.16(a). Fig. 1.1 is a visualization of one of these events.

We take a subset of 1,000 sensors and train the sparsifying bases with 2,000 simulated events. The results in Fig. 6.16(b) show that the learned bases detect on average 0.1 second faster, especially for the more difficult events that are smaller and further away. This saving in time is significant, considering that it takes only about one second for the quake to travel through the network and only 0.5 seconds for the network to be saturated with signals.

6.5 Discussion

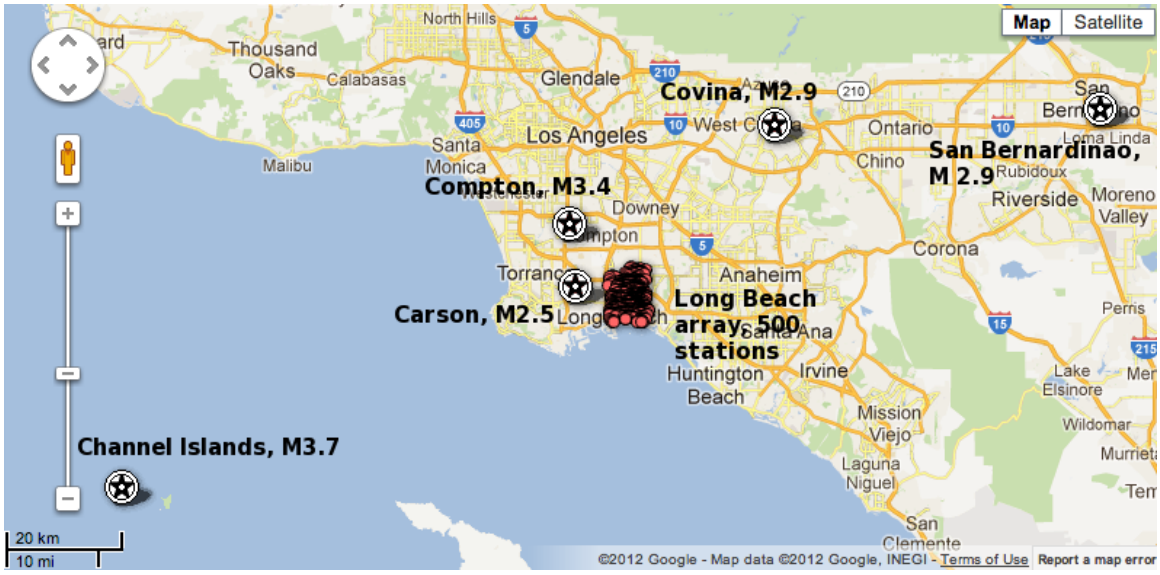
The problem of detection in sensor networks has been widely studied in many application domains; however, little work has been done in the setting of large, dense noisy networks that are feasible with the advance of powerful mobile computing and sensing devices. In this chapter, we explore the simple idea that grouping data from highly correlated sensors improves the signal to noise ratio (and thus detection performance). We also present several scalable algorithms of increasing complexity for efficient grouping that either exploits prior knowledge of the event type or learns

(a) Sensors (red) and events (starred)

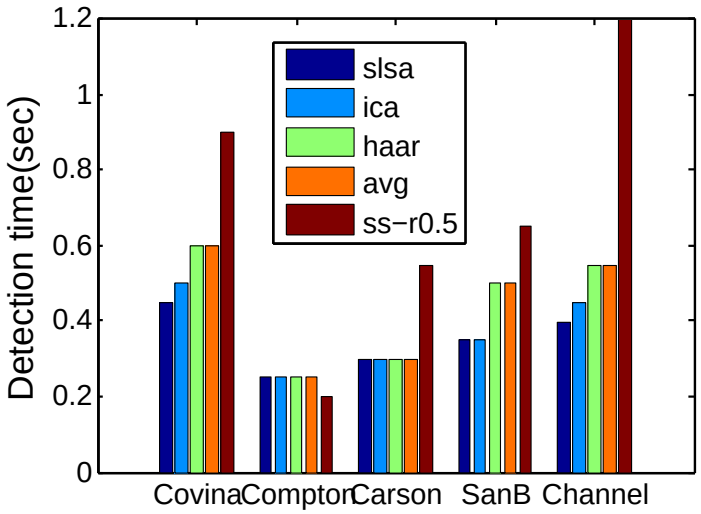


(b) Detection time comparison

Figure 6.15: CSN network. (a) plots the layout of 128 sensors and epicenter of four recorded events. (b) The learned bases detect on average several seconds faster than the baselines under the constraint of, at most, one false alarm per year.



(a) Sensors and events layout



(b) Detection time comparison

Figure 6.16: Long Beach array. (a) shows the layout of 1,000 stations and five recorded events. (b) Under the constraint of, at most, one false alarm per year, the learned bases detect on average 0.1 seconds faster than the baselines, which is significant considering it only takes one second for the seismic wave to travel through the network and only 0.5 seconds for the network to be saturated with signals.

patterns from historic of simulated data. Rigorous analyses of these algorithms are given in terms of both theoretical and experimental performance.

While these algorithms outperform state-of-the art baselines in many real world applications up to thousands of sensors, they, however, only approximate the optimal solution. One can envision other algorithms that approximate the same solution but scale even better. A future work would be to study these algorithms and their relationship to each other. To take it even further, a future research direction would be to establish the limit of detectability in massive noisy networks, much like the Shannon's Theorem on coding theory.

Chapter 7

Conclusions

This chapter summarizes the contributions of the thesis and discusses possible directions for future work on sensor networks for geospatial event detection.

7.1 Summary

Advances in low-cost sensor hardware and infrastructure are making it possible to deploy large dense sensor networks that could rapidly detect important geospatial events such as earthquakes and hazardous radiation. In this thesis, I have presented methodologies for implementing and evaluating dense noisy networks for event detection in terms of speed and accuracy.

In a general sense, several parameters influence the speed and reliability of detection, including the number and quality of sensors; communication capabilities between each sensor and a central server; and the speed and magnitude of event propagation. Motivated by early warning applications, I considered how rapidly and reliably events can be detected when all measurements are available at the server (the centralized setting), and when only binary messages may be sent to the server (the decentralized setting). I provided asymptotic and theoretical analyses that quantify and bound the detection performance in terms of key network and event parameters.

To ensure detection accuracy, I presented a unifying Bayesian framework for estimating event critical parameters (location, starting time, magnitude) under several event and network configurations, taking into accounts possible network and event dynamics. These dynamics range from the most common scenario with stationary event and stationary sensors to variations including traveling events, mobile sensors, and multiple events. To solve the common problem of selecting priors when the event parameters are continuous and unbounded, I introduced an integrated Bayesian approach that constructs the priors from classical estimates. The effectiveness of the algorithms was validated through extensive experimentation with simulated and real measurements. These studies serve to provide a unifying tool for solving the parameter estimation problem of geospatial event under most circumstances.

The recent development of *participatory networks* that consist of sensor-equipped devices operated by the public leads to the discussion of scalability. I studied the detection problem in massive, noisy, participatory networks and presented several scalable geospatial data aggregation algorithms of increasing complexity based on a distributed detection scheme. I showed how learning correlations among sensors at the network layer can improve upon standard techniques for spatial aggregation. Practical issues are discussed, and the approach is demonstrated on four real-world networks. This study demonstrates the great potential of machine learning for revealing hidden patterns among sensors that are often not well captured by models. It also opens up possible future directions for parameter estimation research.

7.2 Future Work

7.2.1 Richer Event Model

Point source assumption. All studies presented in this thesis follow the point source assumption; that is, the event initiates at a single point in space and time. This assumption not only simplifies the event model but is also an appropriate approximation for some event types, e.g. a dirty bomb scenario. However, in some applications this may be an over simplification. For example, the air pollution caused by traffic often does not originate from a single point, but from a line along the highway; the rupture of an earthquake is frequently caused by fracture propagation along a fault plane. In both cases, the event is better modeled as a line rather than a single point. A richer geometric model of event initiation (e.g. line, rectangle) can be easily incorporated into the Bayesian framework with only slight increase in computational load. However, the design of the likelihood function relies on a more complicated model of event manifestation H. How to correctly model this function H for different events is the biggest challenge.

Additive intensity assumption. Without much modification to existing models, event initiation can also be modeled as a collection point sources, or *point cloud*. A point cloud model assumes that independent events can occur closely in space-time. It is an example of the multiple event model that we discussed in Chapter 5. A major challenge with this model is to correctly compute the additive impact intensity at a point in space and time from multiple point initiations. For some events, such as radiation, the additive intensity is simply the summation of expected intensities from all point initiations. For other event types, such as earthquakes, the effective additive intensity is not always the direct summation, but is the result of complex interference from multiple wavefronts. A good understanding of the interactions between multiple events' manifestations is critical for estimating event parameters both for the multiple event case and for the point cloud model.

7.2.2 Analytics

Mixed sensor data fusion. While traditional sensor networks are mostly homogeneous (i.e. consist of only one type of sensor), the new generation of sensor networks are often heterogeneous (i.e. consist of more than one type of sensor). As an example, the Community Seismic Network (CSN) [1] employs two types of sensors: the 16-bit USB standalone accelerometer *Phidget* and the 12-bit smartphone built-in accelerometer *Android*. It is no surprise that during an event, the *Phidget* sensors produce more accurate measurements than those from the *Android* sensors. The varying quality of different types of sensors should be taken into consideration when their measurements are combined at a fusion center. The difference in reliability exists not only between different types of sensors, but also among the same sensor type due to environmental noise. An accelerometer placed on a desk experiences on average more variation in ambient noise than one that is secured on the concrete floor in a basement. The sensor reliability can be computed by asking the question: "What is the probability that sensor s measures an acceleration Y given that there is and is not an event". This is equivalent to computing the conditional probability $\mathbb{P}[y = Y|E = 0]$ and $\mathbb{P}[y = Y|E = 1]$. This reliability measure can change with the time of the day/month/season. It requires further work to properly incorporate this measure into the Bayesian calculation to account for different sensor quality.

Machine learning for parameter estimation. It was shown in Sec. 6.4 that sensor correlations can be learned from historical data to improve detection performance. In fact, learning from such data can reveal many details on the physics of the event manifestation, details that are often not well captured by models. The improvement in system performance with machine learning is not limited to detection. The work on *PreSEIS* by Bosë *et al.* shows that a neural network trained with historical recordings can successfully estimate earthquake parameters such as hypocenter and moment magnitude in real time [10]. Other existing techniques that may achieve similar results include support vector machine, deep learning, . . . , etc.. These types of model-free, non-parametric techniques have great potentials in learning the highly complex patterns of geospatial events that are difficult to model. The questions that remain are whether a technique can be generalized for other types of events and how to construct the training data set for a rare event.

7.2.3 Data privacy

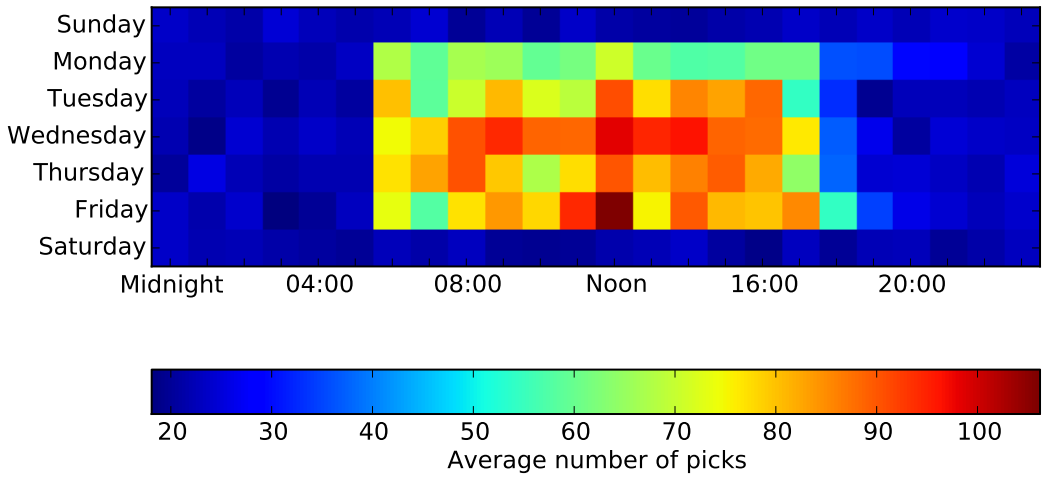
As more and more networks employ human and/or smartphones as sensors, the privacy issue becomes increasingly more important. As a motivating example of data privacy, Fig. 7.1 reveals two plots showing the same type of accelerometer placed at two different locations — (a) office and (b)

[1] CSN: `http://csn.caltech.edu/`

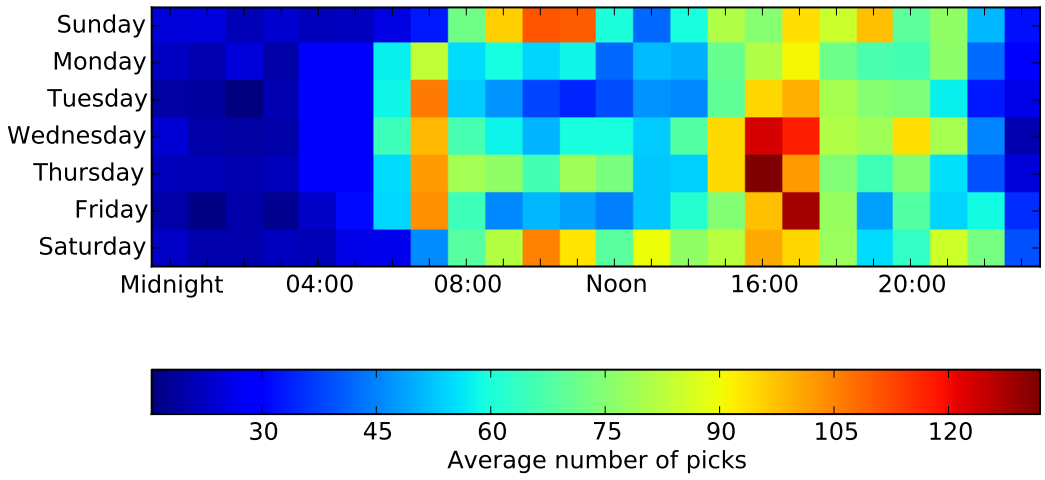
residence. From the aggregated measurements, the sensor owner's daily activity pattern is easily visible. For example, from Fig. 7.1(b), one can easily assess that the probability is very low for someone to be at home between 9am - 1pm on a Tuesday.

As we collect and publish sensor network data for scientific and educational purpose, it is of great importance to make sure that no one can piece together such sensitive information about sensor owners. It is also important to ensure that useful information can be derived from the published data set. To achieve a balance between useful information and data privacy, we may only want to publish randomized or aggregated data. For example, the published lat-lon sensor locations can be slightly corrupted with zero-mean Gaussian noise such that the exact location is only accurate to a certain resolution. This idea is currently implemented on the CSN's sensor activity map such that the activities from two houses on the same block cannot be distinguished by a map viewer.

A more rigorous approach to this problem is *differential privacy*. Here, the goal is to maximize the accuracy of certain queries from data while minimizing the chances of identifying individual records. The core concept of differential privacy is ϵ-*differential privacy*, which tradeoffs utility and privacy via the ϵ term. More discussions on ϵ-differential privacy, as well as other variations of differential privacy, can be found in [31] and many other previous work in the literature.



(a) Office sensor



(b) Residential sensor

Figure 7.1: These graphs show the average number of picks at different times of day, and on different days of the week for an office and a residential sensor. The sensor measurements clearly reveal the activity patterns of the sensor owner

Appendices

Appendix A

A.1 Proof of Lemma 1

Proof. Let $F(x)$ be the cumulative density function of a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$. Without loss of generality, let $\mu = 0$, then $G(x)$ becomes

$$\begin{aligned} G(x) &= \int_{x-\sigma}^{x+\sigma} f(x)\,\mathrm{d}x \\ &= F(x+\sigma) - F(x-\sigma) \\ &= \frac{1}{2}\left[erf\left(\frac{x+\sigma}{\sqrt{2\sigma^2}}\right) - erf\left(\frac{x-\sigma}{\sqrt{2\sigma^2}}\right)\right] \end{aligned}$$

Lower bound. We first prove the lower bound. Let $L(x)$ be the lower bound

$$L(x) = erf\left(\frac{1}{\sqrt{2}}\right) e^{-\frac{x^2}{2\sigma^2}}$$

Let $H(x)$ be the difference between $G(x)$ and the lower bound $L(x)$

$$H(x) = \frac{1}{2}\left[erf\left(\frac{x+\sigma}{\sqrt{2\sigma^2}}\right) - erf\left(\frac{x-\sigma}{\sqrt{2\sigma^2}}\right)\right] - erf\left(\frac{1}{\sqrt{2}}\right) e^{-\frac{x^2}{2\sigma^2}}$$

Because both $G(x)$ and $L(x)$ are symmetric around 0, it suffices to show that $H(x) \geq 0, \forall x \geq 0$ for the lower bound to hold. To show this, we first check the value of $H(x)$ when x is at the extreme. When $x = 0$, we have

$$\begin{aligned} H(0) &= \frac{1}{2}\left[erf\left(\frac{1}{\sqrt{2}}\right) - erf\left(\frac{-1}{\sqrt{2}}\right)\right] - erf\left(\frac{1}{\sqrt{2}}\right) \\ &= erf\left(\frac{1}{\sqrt{2}}\right) - erf\left(\frac{1}{\sqrt{2}}\right) = 0 \end{aligned} \tag{A.1}$$

and when $x = \infty$, we have

$$H(\infty) = \frac{1}{2}\left[erf\left(\infty\right) - erf\left(\infty\right)\right] - 0 = 0 \tag{A.2}$$

Using the *FindRoot* command in *Mathematica* we can confirm that

$$H(x) \neq 0, \forall 0 < x < \infty \tag{A.3}$$

Given that $H(x)$ is smooth and continuous (i.e. $H(x)$ is differentiable everywhere) and $H(x)$ does not cross 0 for all $0 < x < \infty$, if $H(x)$ is increasing just after x moves away from 0, i.e. if $H(x) > 0$ for $0 < x \ll 1$, then we have $H(x) \geq 0, \forall x \geq 0$. To show this, we take the derivative of $H(x)$

$$\begin{aligned} H'(x) &= \frac{-1}{\sigma\sqrt{2\pi e}} e^{-\frac{x^2}{2\sigma^2}} \left(e^{x/\sigma} - e^{-x/\sigma}\right) + \frac{x}{\sigma^2} erf\left(\frac{1}{\sqrt{2}}\right) e^{-\frac{x^2}{2\sigma^2}} \\ &= e^{-\frac{x^2}{2\sigma^2}} \left[\frac{-1}{\sigma\sqrt{2\pi e}} \left(e^{x/\sigma} - e^{-x/\sigma}\right) + \frac{x}{\sigma^2} erf\left(\frac{1}{\sqrt{2}}\right)\right] \end{aligned}$$

Let the term inside the bracket be $h(x)$. Since $e^{-\frac{x^2}{2\sigma^2}} \geq 0$, we only need to show that $h(x)$ is positive when $0 < x \ll 1$ for the statement that $H(x)$ is positive when $0 < x \ll 1$ to be true. We do this by linearizing $h(x)$ around 0 to get the linearized $\bar{h(x)}$

$$\begin{aligned} \bar{h(x)} = h'(0)x &= \left(\frac{-2}{\sigma^2\sqrt{2\pi e}} + \frac{erf\left(\frac{1}{\sqrt{2}}\right)}{\sigma^2}\right) x \\ &= \frac{x}{\sigma^2}\left(erf\left(\frac{1}{\sqrt{2}}\right) - \frac{2}{\sqrt{2\pi e}}\right) \approx \frac{1.1663}{\sigma^2} x \end{aligned}$$

It is clear that $\bar{h(x)} > 0$ when $0 < x \ll 1$. It follows that $H(x) > 0$ when $0 < x \ll 1$. Given this and that $H(x)$ is continuous and smooth, with Eq. (A.1), Eq. (A.2) and Eq. (A.3), we have $G(x) > L(x)$. This concludes the proof for lower bound.

Upper bound. Next we prove the upper bound in a similar way. Let $U(x)$ be the upper bound,

$$U(x) = erf\left(\frac{1}{\sqrt{2}}\right) e^{-\frac{x^2}{4\sigma^2}}$$

Let $H(x)$ be the difference between $G(x)$ and the upper bound $U(x)$

$$H(x) = erf\left(\frac{1}{\sqrt{2}}\right) e^{-\frac{x^2}{4\sigma^2}} - \frac{1}{2}\left[erf\left(\frac{x+\sigma}{\sqrt{2\sigma^2}}\right) - erf\left(\frac{x-\sigma}{\sqrt{2\sigma^2}}\right)\right]$$

Because both $G(x)$ and $U(x)$ are symmetric around 0, it suffices to show that $H(x) \geq 0, \forall x \geq 0$ for the lower bound to hold. To show this, we first check the value of $H(x)$ when x is at the extreme.

When $x = 0$, we have

$$\begin{aligned} H(0) &= erf\left(\frac{1}{\sqrt{2}}\right) - \frac{1}{2}\left[erf\left(\frac{1}{\sqrt{2}}\right) - erf\left(\frac{-1}{\sqrt{2}}\right)\right] \\ &= erf\left(\frac{1}{\sqrt{2}}\right) - erf\left(\frac{1}{\sqrt{2}}\right) = 0 \end{aligned} \tag{A.4}$$

and when $x = \infty$, we have

$$H(\infty) = 0 - \frac{1}{2}\left[erf\left(\infty\right) - erf\left(\infty\right)\right] = 0 \tag{A.5}$$

Using the *FindRoot* command in *Mathematica* we can confirm that

$$H(x) \neq 0, \forall 0 < x < \infty \tag{A.6}$$

Given that $H(x)$ is smooth and continuous (i.e. $H(x)$ is differentiable everywhere) and $H(x)$ does not cross 0, for all $0 < x < \infty$, if $H(x)$ is increasing just after x moves away from 0, i.e. if $H(x) > 0$ when $0 < x \ll 1$, then we have $H(x) \geq 0, \forall x \geq 0$. To show this, we take the derivative of $H(x)$

$$\begin{aligned} H'(x) &= -\frac{x}{2\sigma^2} erf\left(\frac{1}{\sqrt{2}}\right) e^{-\frac{x^2}{4\sigma^2}} + \frac{1}{\sigma\sqrt{2\pi e}} e^{-\frac{x^2}{2\sigma^2}} \left(e^{x/\sigma} - e^{-x/\sigma}\right) \\ &= e^{-\frac{x^2}{4\sigma^2}} \left[-\frac{x}{2\sigma^2} erf\left(\frac{1}{\sqrt{2}}\right) + e^{-\frac{x^2}{4\sigma^2}} \frac{1}{\sigma\sqrt{2\pi e}} \left(e^{x/\sigma} - e^{-x/\sigma}\right)\right] \end{aligned} \tag{A.7}$$

Let the term inside the bracket be $h(x)$. Since $e^{-\frac{x^2}{4\sigma^2}} \geq 0$, we only need to show that $h(x)$ is positive for $0 < x \ll 1$. We do this by linearizing $h(x)$ around 0 to get the linearized $\bar{h(x)}$

$$\begin{aligned} \bar{h(x)} = h'(0)x &= \left(-\frac{erf\left(\frac{1}{\sqrt{2}}\right)}{2\sigma^2} + \frac{2}{\sigma^2\sqrt{2\pi e}}\right) x \\ &= \frac{x}{2\sigma^2}\left(\frac{4}{\sqrt{2\pi e}} - erf\left(\frac{1}{\sqrt{2}}\right)\right) \approx \frac{0.142597}{\sigma^2} x \end{aligned} \tag{A.8}$$

It is clear that $\bar{h(x)} > 0$ when $0 < x \ll 1$. It follows that $H(x) > 0$ when $0 < x \ll 1$. Given this and that $H(x)$ is continuous and smooth, with Eq. (A.4), Eq. (A.5) and Eq. (A.6), we have $U(x) > G(x)$. This concludes the proof for the upper bound. Fig. A.1 visualizes the lower bound and upper bound with respect to the exact value as a function of x.

□

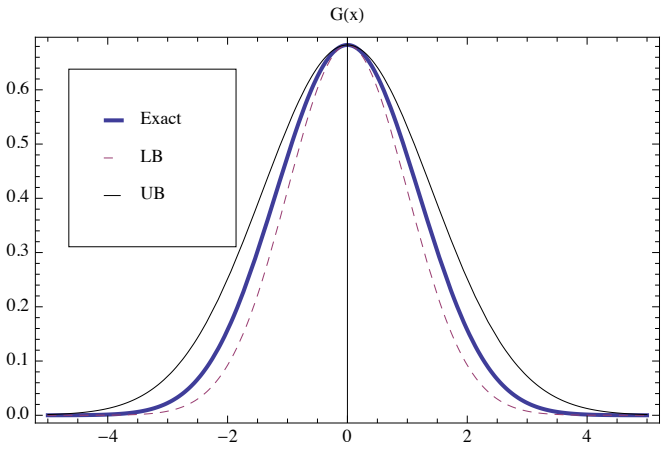


Figure A.1: Visualization of Lemma 1 with $\mu = 0$ and $\sigma = 1$.

Bibliography

[1] Earthquake early warning report for tohoku 2011. http://www.jma.go.jp/jma/press/1104/28b/eew_hyouka_2.pdf.

[2] ishake project. http://ishakeberkeley.appspot.com/.

[3] P. Huggins S. Labov K. Nelson Y. Yao J. Tuszynski J. Briggs D. Cohen A. Dubrawski, S. Ray. Machine learning approach to support determination of nuclear threat. In *IEEE Symposium on Radiation Measurements and Applications*, 2012.

[4] Japan Meteorological Agency. Reference on the techniques for earthquake early warning. http://www.seisvol.kishou.go.jp/eq/EEW/kaisetsu/Whats_EEW/reference.pdf, 2010. Accessed: 2013-03-26.

[5] M.S Arulampalam, S Maskell, N Gordon, and T Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.

[6] Javed Aslam, Zack Butler, Florin Constantin, Valentino Crespi, George Cybenko, and Daniela Rus. Tracking a moving object with a binary sensor network. In *SenSys '03: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*. ACM Request Permissions, November 2003.

[7] Gail M Atkinson and David J Wald. "Did You Feel It?" Intensity Data: A Surprisingly Good Measure of Earthquake Ground Motion. *Seismological Research Letters*, 2007.

[8] Japan Nuclear Regulation Authority. Monitoring information of environmental radioactivity level. http://radioactivity.nsr.go.jp/ja/contents/8000/7312/24/20130318air.pdf, 2013.

[9] M Böse, E Hauksson, K Solanki, H. Kanamori, and T H Heaton. Real-time testing of the on-site warning algorithm in southern California and its performance during the July 29 2008 M w5.4 Chino Hills earthquake. *Geophysical Research Letters*, 36(3):L00B03, February 2009.

[10] Maren Böse, Friedemann Wenzel, and Mustafa Erdik. PreSEIS: A neural network-based approach to earthquake early warning for finite faults. *Bulletin of the Seismological Society of America*, 98(1):366–382, 2008.

[11] S M Brennan, A M Mielke, D C Torney, and A B Maccabe. Radiation detection with distributed sensor networks. *Computer*, 37(8):57–59, 2004.

[12] Peter Bühlmann and Sara van de Geer. *Statistics for High-Dimensional Data*. Methods, Theory and Applications. Springer, June 2011.

[13] E J Candes and M B Wakin. An Introduction To Compressive Sampling. *IEEE Signal Processing Magazine*, 25(2):21–30, 2008.

[14] Aranya Chakrabortty and Clyde F Martin. Optimal sensor placement for parametric model identification of electrical networks, Part II: Estimation under output feedback. In *49th IEEE Conference on Decision and Control (CDC)*, pages 5810–5815, 2010.

[15] K. Mani Chandy, Julian Bunn, and Annie H. Liu. Models and Algorithms for Radiation Detection. In *Modeling and Simulation Workshop for Homeland Security*, pages 1–6. Caltech, March 2010.

[16] M Chandy, C Pilotto, and R McLean. Networked sensing systems for detecting people carrying radioactive material. *International Conference on Networked Sensing Systems*, Jan 2008.

[17] S S Chen, D L Donoho, and M A Saunders. Atomic Decomposition by Basis Pursuit. *SIAM Review*, 2001.

[18] Wei-Peng Chen, Jennifer C Hou, and Lui Sha. Dynamic Clustering for Acoustic Target Tracking in Wireless Sensor Networks. In *Proceedings of the 11th IEEE International Conference on Network Protocols (ICNP'03)*.

[19] X Chen, Y Qi, B Bai, Q Lin, and J G Carbonell. Sparse latent semantic analysis. *NIPS Workshop*, 2010.

[20] J Cheng, M Xie, R Chen, and F Roberts. A mobile sensor network for the surveillance of nuclear materials in metropolitan areas. *DIMACS Technical Report*, 2009.

[21] J Chin, D Yau, N Rao, Y Yang, and C Ma. Accurate localization of low-level radioactive source under noise and measurement errors. *Proceedings of the 6th ACM Conference on Embedded Network Sensor*, Jan 2008.

[22] Robert W. Clayton, Thomas Heaton, Mani Chandy, Andreas Krause, Monica Kohler, Julian Bunn, Richard Guy, Michael Olson, Mathew Faulkner, MingHei Cheng, Leif Strand, Rishi

Chandy, Daniel Obenshain, Annie Liu, and Michael Aivazis. Community Seismic Network. *Annals of Geophysics*, 54(6), January 2012.

[23] Elizabeth S Cochran, Jesse F Lawrence, Carl Christensen, and Ravi S Jakka. The quake-catcher network: Citizen science expanding seismic horizons. *Seismological Research Letters*, 80(1):26–30, 2009.

[24] G Cua and T Heaton. The Virtual Seismologist (VS) method: A Bayesian approach to earthquake early warning. *Earthquake Early Warning Systems*, pages 97–132, 2007.

[25] Georgia Cua, Michael Fischer, Thomas Heaton, and Stefan Wiemer. Real-time Performance of the Virtual Seismologist Earthquake Early Warning Algorithm in Southern California. *Seismological Research Letters*, 80(5):740, September 2009.

[26] Georgia B Cua. *Creating the Virtual Seismologist: developments in ground motion characterization and seismic early warning*. PhD thesis, thesis.library.caltech.edu, 2005.

[27] G Davis, S Mallat, and M Avellaneda. Adaptive greedy approximations. *Constructive Approximation*, 13(1):57–98, March 1997.

[28] Mark De Berg, Otfried Cheong, and Marc Van Kreveld. Computational geometry: algorithms and applications. In *Computational Geometry: Algorithms and Applications*. Springer, 2008.

[29] A. Doucet, N. de Freitas, N. Gordon, and A. Smith. *Sequential Monte Carlo Methods in Practice*. Statistics for Engineering and Information Science. Springer, 2001.

[30] Devdatt Dubhashi and Alessandro Panconesi. Concentration of Measure for the Analysis of Randomized Algorithms, 1st edition. *Concentration of Measure for the Analysis of Randomized Algorithms, 1st edition*, June 2009.

[31] Cynthia Dwork. Differential privacy. *Automata, languages and programming*, pages 1–12, 2006.

[32] Jakob Eriksson, Lewis Girod, Bret Hull, Ryan Newton, Samuel Madden, and Hari Balakrishnan. The pothole patrol: using a mobile sensor network for road surface monitoring. In *MobiSys '08: Proceeding of The 6th International Conference on Mobile Systems, Applications, and Services*. ACM Request Permissions, June 2008.

[33] Mari Ervasti, Shideh Dashti, Jack Reilly, Jonathan D Bray, Alexandre Bayen, and Steven Glaser. iShake: mobile phones as seismic sensors – user study findings. In *MUM '11: Proceedings of the 10th International Conference on Mobile and Ubiquitous Multimedia*. ACM Request Permissions, December 2011.

[34] M Faulkner, M Olson, R Chandy, J Krause, K M Chandy, and A Krause. The next big one: Detecting earthquakes and other rare events from community-based sensors. In *Proceedings of the 10th International Conference on Information Processing in Sensor Networks (IPSN)*, 2011.

[35] Matthew Faulkner, Annie H. Liu, and Andreas Krause. A Fresh Perspective: Learning to Sparsify for Detection in Massive Noisy Sensor Networks. In *Proceedings of The 12th International Conference on Information Processing in Sensor Networks (IPSN)*, 2013.

[36] Raghu Ganti, Iqbal Mohomed, Ramya Raghavendra, and Anand Ranganathan. *Analysis of Data from a Taxi Cab Participatory Sensor Network*, volume 104. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[37] Subir Kumar Ghosh. Approximation algorithms for art gallery problems in polygons. *Discrete Applied Mathematics*, 158(6):718–722, March 2010.

[38] J.C. Gower and G.B. Dijksterhuis. *Procrustes Problems*. Oxford Statistical Science Series. OUP Oxford, 2004.

[39] A Gunatilaka, B Ristic, and R Gailis. On localisation of a radiological point source. *Information, Decision and Control*, Jan 2007.

[40] A Gunatilaka, B Ristic, and R Gailis. Radiological source localisation. *dspace.dsto.defence.gov.au*, Jan 2007.

[41] D Hochbaum and B Fishbain. Nuclear threat detection with mobile distributed sensor networks. *Annals of Operations Research*, Jan 2009.

[42] Dorit S Hochbaum and Wolfgang Maass. Approximation schemes for covering and packing problems in image processing and VLSI. *Journal of the ACM*, 32(1):130–136, January 1985.

[43] Shigeki Horiuchi, Hiroaki Negishi, Kana Abe, Aya Kamimura, and Yukio Fujinawa. An automatic processing system for broadcasting earthquake alarms. *Bulletin of the Seismological Society of America*, 95(2):708–718, 2005.

[44] J Howse, L Ticknor, and K Muske. Least squares estimation techniques for position tracking of radioactive sources. *Automatica*, Jan 2001.

[45] A Hyvärinen. Independent Component Analysis: Algorithms and Applications. *Neural Networks*, 2000.

[46] Aapo Hyvärinen, Juha Karhunen, and Erkki Oja. *Independent Component Analysis*. Wiley-Interscience, June 2001.

[47] M. Faulkner A. H. Liu M. Olson J. J. Bunn, K. M. Chandy. Sense and response systems for crisis management. In S.K. Das, K. Kant, and N. Zhang, editors, *Handbook on Securing Cyber-Physical Critical Infrastructure*. Elsevier Science & Technology, 2012.

[48] T T Warnock R J Nemzek J S Dreicer, D C Torney. Distributed sensor networks for detection of mobile radioactive sources. *IEEE Transactions on Nuclear Science*, 5(4):1693–1699, Sep 2004.

[49] K Jarman, L Smith, and D Carlson. Sequential probability ratio test for long-term radiation monitoring. *2003 IEEE Nuclear Science Symposium Conference*, Jan 2003.

[50] Akio Katsumata. Recursive digital filter with frequency response of a mechanical seismograph. Technical report, Seismology and Volcanology Research Department, Meteorological Research Institute, 2008.

[51] Nina Köhler, Georgia Cua, Friedemann Wenzel, and Maren Böse. Rapid source parameter estimations of southern California earthquakes using PreSEIS. *Seismological Research Letters*, 80(5):748–754, 2009.

[52] A Krause, A Singh, and C Guestrin. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *The Journal of Machine Learning Research*, 2008.

[53] Akshay Krishnamurthy, James Sharpnack, and Aarti Singh. Detecting Activations over Graphs using Spanning Tree Wavelet Bases. *arXiv.org*, stat.ML, June 2012.

[54] Barry R Lienert, E Berg, and L Neil Frazer. HYPOCENTER: An earthquake location method using centered, scaled, and adaptively damped least squares. *Bulletin of the Seismological Society of America*, 1986.

[55] Chih-Yu Lin, Wen-Chih Peng, and Yu-Chee Tseng. Efficient in-network moving object tracking in wireless sensor networks. *IEEE Transactions on Mobile Computing*, 5(8):1044–1056, August 2006.

[56] Annie Liu, Michael Olson, Julian Bunn, and K. Mani Chandy. Towards a discipline of geospatial distributed event based systems. In *DEBS '12: Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems*, July 2012.

[57] Annie H. Liu. Simulation and implementation of distributed sensor network for radiation detection. Master's thesis, California Institute of Technology.

[58] Annie H. Liu, Julian Bunn, and K. Mani Chandy. An analysis of data fusion for radiation detection and localization. In *Proceedings of the 13th Conference on Information Fusion (FUSION)*, pages 1–8, 2010.

[59] Annie H. Liu, Julian Bunn, and K. Mani Chandy. Sensor networks for the detection and tracking of radiation and other threats in cities. In *Proceedings of the 10th International Conference on Information Processing in Sensor Networks (IPSN)*, pages 1–12, 2011.

[60] Jun S. Liu and Rong Chen. Sequential monte carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93:1032–1044, 1998.

[61] David J C MacKay. *Bayesian methods for adaptive models* . PhD thesis, California Institute of Technology, 1992.

[62] Waseem A Malik, Nuno C Martins, and Ananthram Swami. Optimal sensor placement for intruder detection: New design principles. In *Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on*, pages 1537–1543, 2011.

[63] M Morelande, B Ristic, and A Gunatilaka. Detection and parameter estimation of multiple radioactive sources. In *10th International Conference on Information Fusion*, pages 1–7, 2007.

[64] M Morelande and A Skvortsov. Radiation field estimation using a gaussian mixture. *International Conference on Information Fusion*, 2009.

[65] C. Musso, N. Oudjane, and FranÇ. Legland. Improving regularized particle filters. In A. Doucet, N. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo Methods in Practice. New York*, number 12, pages 247–271. Statistics for Engineering and Information Science, 2001.

[66] Daniel B. Neill. Fast subset scan for spatial pattern detection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(2), 2012.

[67] G Nemhauser, L Wolsey, and M Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 1978.

[68] S E J Nippress, A Rietbrock, and A E Heath. Optimized automatic pickers: application to the ANCORP data set. *International Geophysical Journal*, March 2010.

[69] Chang Kook Oh, James L Beck, and Masumi Yamada. Bayesian Learning Using Automatic Relevance Determination Prior with an Application to Earthquake Early Warning. *Journal of Engineering Mechanics*, 134(12):1013–1020, 2008.

[70] Michael Olson, Annie Liu, Matthew Faulkner, and K. Mani Chandy. Rapid detection of rare geospatial events: earthquake warning applications. In *DEBS '11: Proceedings of The 5th ACM International Conference on Distributed Event-Based System*. ACM Request Permissions, July 2011.

[71] Gary L Pavlis and John R Booker. The mixed discrete-continuous inverse problem: Application to the simultaneous determination of earthquake hypocenters and velocity structure. *Journal of Geophysical Research*, 85(B9):4801, 1980.

[72] Gary L. Pavlis, Frank Vernon, Danny Harvey, and Dan Quinlan. The generalized earthquake-location (genloc) package: an earthquake-location library. *Computational Geoscience*, 30(9-10):1079–1091, November 2004.

[73] N S V Rao, Jren-Chit Chin, D K Y Yau, and C Y T Ma. Localization leads to improved distributed detection under non-smooth distributions. In *13th Conference on Information Fusion*, pages 1–8, 2010.

[74] N S V Rao, C W Glover, M Shankar, Jren-Chit Chin, D K Y Yau, C Y T Ma, Yong Yang, and S Sahni. Improved SPRT detection using localization with application to radiation sources. In *FUSION '09. 12th International Conference on Information Fusion*, pages 633–640, 2009.

[75] N S V Rao, M Shankar, Jren-Chit Chin, D K Y Yau, C Y T Ma, Yong Yang, J C Hou, Xiaochun Xu, and S Sahni. Localization under random measurements with application to radiation sources. In *11th International Conference on Information Fusion*, pages 1–8, 2008.

[76] N S V Rao, M Shankar, Jren-Chit Chin, D K Y Yau, S Srivathsan, S S Iyengar, Yong Yang, and J C Hou. Identification of Low-Level Point Radiation Sources Using a Sensor Network. In *IPSN '08. International Conference on Information Processing in Sensor Networks*, pages 493–504, 2008.

[77] B Ristic, A Gunatilaka, and M Rutten. An information gain driven search for a radioactive point source. In *10th International Conference on Information Fusion*, pages 1–8, 2007.

[78] B Ristic, M Morelande, and A Gunatilaka. A controlled search for radioactive point sources. In *11th International Conference on Information Fusion*, pages 1–5, 2008.

[79] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes Twitter users: real-time event detection by social sensors. *Proceedings of the 19th International Conference on World Wide Web*, pages 851–860, 2010.

[80] Malcolm Sambridge and Kerry Gallagher. Earthquake hypocenter location using genetic algorithms. *Bulletin of the Seismological Society of America*, 83(5):1467–1491, October 1993.

[81] G. Shmueli and H. Burkom. Statistical challenges facing early outbreak detection in biosurveillance. *Technometrics*, 52(1):39–51, 2010.

[82] Aarti Singh, Robert Nowak, and Robert Calderbank. Detecting Weak but Hierarchically-Structured Patterns in Networks. In *The 13th International Conf. on Artificial Intelligence and Statistics (AISTATS)*, September 2010.

[83] Yu-Chee Tseng, Sheng-Po Kuo, Hung-Wei Lee, and Chi-Fu Huang. Location tracking in a wireless sensor network by mobile agents and its data fusion strategies. In *IPSN'03: Proceedings of the 2nd International Conference on Information Processing in Sensor Networks*. Springer-Verlag, April 2003.

[84] C. Vincent Zhou, C. Leckie, and S. Karunasekera. Decentralized multi-dimensional alert correlation for collaborative intrusion detection. *Journal of Network and Computer Applications*, 32(5):1106–1123, 2009.

[85] J Weng and B S Lee. Event detection in Twitter. In *Proceedings of the 5th International AAAI Conference*, 2011.

[86] David Bruce Wilson. Generating random spanning trees more quickly than the cover time. In *STOC '96: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. ACM Request Permissions, July 1996.

[87] M Wu, A Liu, and K M Chandy. Virtual environments for developing strategies for interdicting terrorists carrying dirty bombs. *International Society of Crisis Response and Management Conference*, pages 1–5, 2008.

[88] Yih-Min Wu and Hiroo Kanamori. Development of an Earthquake Early Warning System Using Real-Time Strong Motion Signals. *Sensors*, 8(1):1–9, January 2008.

[89] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *ACM Computing Surveys*, 38(4):13–es, December 2006.

[90] W Zhang, L Zhang, Y Ding, and T Miyaki. Mobile Sensing in Metropolitan Area: Case Study in Beijing. *ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2011.

[91] Shiyu Zhao, Ben M Chen, and Tong H Lee. Optimal Sensor Placement for Target Localization and Tracking in 2D and 3D. *arXiv.org*, math.OC, October 2012.

[92] K Ziock and W Goldstein. The lost source, varying backgrounds and why bigger may not be better. *AIP Conference Proceedings*, 2002.